PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-049585

(43)Date of publication of application : 15.02.2002

(51)Int.Cl. G06F 15/00
G06F 9/46
G06F 13/00
G06F 17/30

(21)Application number : 2001-129922

(22)Date of filing : 26.04.2001

(71)Applicant : MICROSOFT CORP

(72)Inventor : BURD GARY S
COOPER KENNETH B
ANDERS MARK T
GUTHRIE SCOTT D
EBBO DAVID S
PETERS TED A
MILLET STEPHEN J

(30)Priority

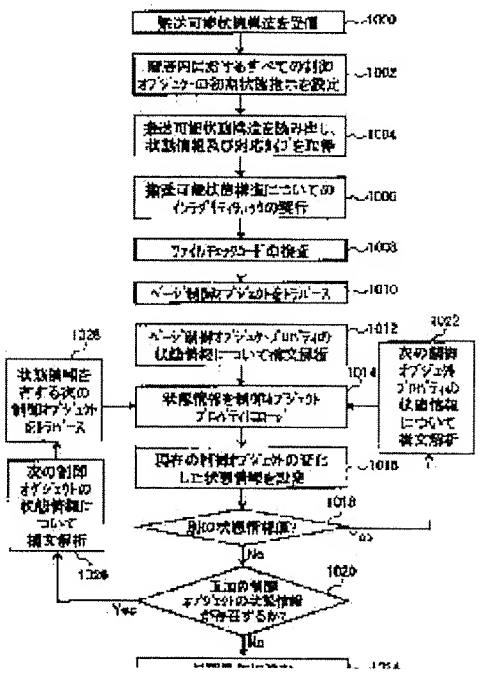
Priority number : 2000 574144 Priority date : 18.05.2000 Priority country : US

(54) STATE MANAGEMENT OF SERVER-SIDE CONTROL OBJECT

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a development framework which can dynamically produce and process a Web page with a minimum program.

SOLUTION: The states of one or more server control objects are managed by means of a transportable state structure that reciprocates between a Web server and a client. In regard to the state information of the transportable state structure, the hierarchy information is used by the server to search for a proper server control object out of the hierarchy where the relative sate information should be loaded about properties of the server control objects. Then an integrity code enables



the server to judge whether the transportable state structure is altered while it is reciprocating between the server and client. The state information on one or more server control objects are recorded on the transportable state structure that transmits these state information to the client before a response is sent to the client. The transportable state structure is returned to the server and loaded into the server control object to reset the hierarchy in its precedent state.

LEGAL STATUS

[Date of request for examination]	04.12.2003
[Date of sending the examiner's decision of rejection]	22.08.2006
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	
[Date of registration]	
[Number of appeal against examiner's decision of rejection]	2006-026205
[Date of requesting appeal against examiner's decision of rejection]	20.11.2006
[Date of extinction of right]	

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]
[Claim 1] It is the status management approach of the server side control object corresponding to the client side user interface element built into the web page displayed on a client. The process which creates the server side control object for processing a client side user interface element to a control object hierarchy, The process which receives from a client the condition structure including the status information which shows the status value of said at least one server side control object in said control object hierarchy which can be conveyed, The status management approach of a server side control object including the process which extracts said status information from said condition structure which can be conveyed, and the process which loads the status value to the property of a server side control object when the status value from said status information relates to the server side control object.
[Claim 2] The status-management approach of the server side control object according to claim 1 which includes the process which extracts the status value relevant to the property of said said at least one server side control object from said condition structure which can be conveyed, and the process which recognizes the hierarchy location of said server side control object in said control object hierarchy based on the hierarchy within [which can be conveyed / said] condition structure in said process to extract.
[Claim 3] The status-management approach of a server side control object including the process which directs said server side control object in said control object hierarchy in said process to load based on the hierarchy location identified by discernment processing, the process which traverse the hierarchy location of said server side control object in said control object hierarchy, and the process which store status value in the property of said server side control object according to claim 2.
[Claim 4] Said condition structure which can be conveyed is the status-management approach of the server side control object which includes further the process which reads said receiving integrity code from said condition structure which can be conveyed, the process which calculate a count integrity code from the status information included in said condition structure which can be conveyed, and the process which judge that said receiving integrity code compares with said count integrity code, and said condition structure which can be conveyed is effective, including a receiving integrity code further.
[Claim 5] The process initialized so that the property of said server side control object may be in an initial state, The process which sets up the index which shows that the condition of the property of said server side control object is not changing from said initial state, The status management approach of a server side control object including the process which changes said index so that it may be shown that the condition of the property of said server side control object changed from said initial state, when the status value from said condition structure which can be conveyed is loaded to a property according to claim 1.
[Claim 6] The status management approach of a server side control object according to claim 5 that the property of said server side control object includes further the process which sets up the index which shows not changing from an initial state when said property is data combined with the server side data storage field.
[Claim 7] The status management approach of the server side control object according to claim 5 which

includes further the process which stores the status information of said server side control object in said condition structure for returning to a client which can be conveyed when said index expresses that the status information of said server side control object changed from the initial state as the process which traverses said each server side control object in said control object hierarchy.
[Claim 8] The status management approach of the server side control object according to claim 7 which includes further the process which transmits said condition structure which can be conveyed to a client by the response which has authoring language data which define a web page.
[Claim 9] The status-management approach of the server side control object according to claim 7 which includes further the process which calculates an integrity code from the status information which answers the process which stores said status information and is included in said condition structure which can be conveyed, the process which store said integrity code in said condition structure which can be conveyed, and the process which transmit to a client said condition structure which can be conveyed by the response which has authoring language data which define a web page.
[Claim 10] The status management approach of the server side control object according to claim 1 which includes further the process which generates the authoring code which defines a part of web page [at least] from said server side [at least one] control object of said control object hierarchy.
[Claim 11] It is the web page in which it is defined by the authoring language code, and it is the web page which can be interpreted by the browser which carries out executive operation on the client computer system connected to the web server, said authoring language code is substantially generated by the web server by the status management approach of a server side control object according to claim 10, and said authoring language code includes said condition structure which can be conveyed of storing the status information of said at least one server side control object of said control object hierarchy.
[Claim 12] Reading [computer system] is possible. To a client It is the computer program storage which codes the computer program which carries out executive operation of the computer process which manages the condition of the server side control object corresponding to the client side user interface element built into the web page displayed. The process which said computer process creates said server side control object to a control object hierarchy, and processes said client side user interface element, The process which receives from a client the condition structure including the status information which shows the status value of said at least one server side control object of said control object hierarchy which can be conveyed, A computer program storage including the process which extracts status information from said condition structure which can be conveyed, and the process which loads the status value to the property of said server side control object when the status value from said status information is connected with said server side control object.
[Claim 13] Shape is taken by the computer system by the subcarrier. It is the computer data signal which codes the computer program which performs the computer process which manages the condition of the server side control object corresponding to the client side user interface element built into the web page displayed on a client. The process which said computer process creates said server side control object to a control object hierarchy, and processes said client side user interface element, The process which receives from a client the condition structure including the status information which shows the status value of said at least one server side control object of said control object hierarchy which can be conveyed, A computer data signal including the process which extracts status information from said condition structure which can be conveyed, and the process which loads the status value to the property of said server side control object when the status value from said status information is connected with said server side control object.
[Claim 14] It is created by the control object hierarchy. The computer process which manages the condition of two or more server side control objects corresponding to two or more client side user interface elements built into the web page displayed on a client is set to a computer system. It is the computer program product which codes the computer program to execute. Said computer process The process which receives from a client the condition structure including the status information relevant to said one or more server side control objects of said control object hierarchy which can be conveyed, The process which un-serializes said status information and extracts the hierarchy of the property of status

value, a related property data type, and said server side control object, A computer program product including the process which discovers said server side control object in said control object hierarchy based on said hierarchy, and the process which loads said status value to the property of said server side control object.
[Claim 15] Said computer process is a computer program product according to claim 14 which includes further the process which changes status value so that it may have said related property data type before said process to load.
[Claim 16] The process initialized so that said computer process may be in an initial state about the property of said server side control object, The process which sets up the index which shows that the property of said server side control object is not changing from said initial state, A computer program product including the process which changes said index so that it may be shown that the condition of the property of said server side control object changed from the initial state, when the status value from said condition structure which can be conveyed is loaded to said property according to claim 14.
[Claim 17] Said process to load includes the process which traverses said one or more server side control objects of said control object hierarchy. As opposed to said each server side control object said process to load The process which extracts a property value from said server side control object relevant to the index which shows that the condition of said server side control object changed from the initial state, The computer program product which includes further the process serialized in said condition structure for returning said property value to the client which has a property value from said other server side control objects in said control object hierarchy which can be conveyed.
[Claim 18] The computer program product according to claim 16 in which said process to load includes further the process which extracts a property data type from said server side control object relevant to said property value, and the process serialized in said condition structure for returning said property data type to the client which has the property value of said server side control object which can be conveyed.
[Claim 19] The computer program product according to claim 16 in which said process to load includes further the process which stores the hierarchy about said server side control object of said control object hierarchy in said condition structure for returning to a client which can be conveyed.
[Claim 20] The computer program product according to claim 17 in which said computer process includes further the process which transmits said condition structure which can be conveyed to a client by the response which has authoring language data which define a web page.
[Claim 21] The computer program product according to claim 17 which includes further the process at which said computer process calculates an integrity code from the status information which answered said process to serialize and was stored in said condition structure which can be conveyed, the process which stores said integrity code in said condition structure which can be conveyed, and the process which transmit to a client said condition structure which can be conveyed by the response which has authoring language data which define a web page.
[Claim 22] Said condition structure which can be conveyed is the computer program product which includes further the process which reads said receiving integrity code from said condition structure which can be conveyed, the process which calculate a count integrity code from the status information included in said condition structure which can be conveyed, and the process which judge that said receiving integrity code compares with said count integrity code, and said condition structure which can be conveyed is effective, including a receiving integrity code further.
[Claim 23] It is the handler system by which executive operation is carried out by server computer which processes a request including the condition structure which can be conveyed of storing the status value and the data type from a client computer. He is the hierarchy of the server side control object relevant to [it is created by the handler and] a client side user interface element. With the hierarchy of the server side control object in which a server side [at least one] control object contains a property The handler system containing the loader module which stores in the property of said server side control object the status value received in said condition structure which can be conveyed based on a data type, and the preservation module which stores said property and data type in said condition structure which can be conveyed.

[Claim 24] The handler system according to claim 23 which contains further the read-out module which extracts the data type from status value, a related hierarchy, and the condition structure that can be conveyed, and the traverse module which directs a control object within said control object hierarchy based on said related hierarchy.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] Generally especially this invention relates to management of the condition of the server side control object which processes the client side user interface element of a web page about a web server framework.
[0002]
[Description of the Prior Art] A typical web browser receives data from the web server which specifies the appearance of a web page and basic actuation which are displayed in a client system. As a typical procedure, a user specifies the uniform resource (resource) locator (henceforth "URL") which is the global address of the resource on World Wide Web, and a desired website is accessed. Generally, the vocabulary "a resource" is the data which can be accessed by the program, or a routine. An example of URL is "HYPERLINK "http://www.microsoft.com/ms.htm" http://www.microsoft.com/ms.htm." The 1st part of this example of URL shows the given protocol (for example, "http") used for a communication link. The 2nd part specifies the domain name (for example, "HYPERLINK "http://www.microsoft.com" www.microsoft.com") which shows the whereabouts of a resource. The 3rd part specifies the resource in a domain (for example, file called "ms.htm"). This is followed and they are a browser and HYPERLINK "http://www.microsoft.com" www.microsoft.com. The HTTP (hypertext transport protocol) request relevant to the example of URL for taking out the data relevant to the ms.htm file in a domain is generated. The web server which is acting as the host of the www.microsoft.com site returns reception, the demanded web page, or a resource to a client system by the HTTP response, and displays a HTTP request on a browser.
[0003] The "ms.htm" file of the above-mentioned example contains the static HTML (HyperText Markup Language) code. HTML is a plane (plaintext) text authoring language used for the document (for example, web page) creation on World Wide Web. Since it is such, an HTML file is taken out from a web server and can be displayed on a browser as a web page. A developer can specify the text and list which are displayed on a browser and by which formatting was carried out, form, a table, a hypertext link, an in-line image, voice, and background graphics using HTML, and a graphical experience which it comes to expect while the user is looking at the information from the Internet can be offered. However, an HTML file is a static file which is not supporting dynamic generation of web page contents in essence.
[0004] When displaying dynamic contents, such as change of a stock price, and traffic information, the server side application program for handling the dialogue between usually more complicated client-server is developed. A server side application program processes a HTTP request, and generates the suitable HTML code for transmission of the HTTP response to a client. An example of a HTTP request may contain parameters, such as data from the data or the web based form in an enquiry string. Since it is such, a server side application program can process this parameter, and can generate the HTML code dynamically by the HTTP response to a client. The document containing the suitable HTML code may be generated dynamically, using the sequence of one or more text write-in processings to a memory

device by which formatting was carried out as an example of a server side application program. Then, it is transmitted to a client system by the HTTP response, and the obtained document is displayed on a browser as a web page there.
[0005] Development of a server side application program is a complicated activity as which it is required it is not only well versed in the usual HTML coding used for a web page design, but that it is well versed in pro GURAMUBE six including one or more programming language (for example, C++, Perl, Visual Basic, or Jscript). However, a web page designer is graphic designer or an editor in many cases, and may be inexperienced in a program. Furthermore, if complicated web page development is simplified, the rate of development of new web contents can be gathered by any developers. Generally, a great effort is required and development of a custom-made server side application program is also actually like [which a developer regards as not wanting to often try it]. Therefore, it is desirable to offer the development framework to which a developer can create and process a web page dynamically by the minimum program.
[0006]
[Problem(s) to be Solved by the Invention] One means which makes min the requirements for a program of dynamic web page generation is an active server page (ASP) framework offered by Microsoft Corp. An ASP resource processes typically the HTTP request which specifies an ASP resource as a desired resource, after that, generates the HTML code as a result of the HTTP response to a client, for example, contains Visual Basic or Jscript. Furthermore, a page developer does not require that all components should be written in from a scratch (from a condition without anything), and or the ASP resource was developed beforehand, refer to a third person's client side library component (for example, client side "ACTIVEX" control) for it. However, in the present server side application framework, programming which must manage dynamically client side user interface elements (for example, a text box, a list box, a carbon button, a hyperlink, an image, voice, etc.) within server side application needs a still advanced programming technique and a considerable effort. It is that an unsolved technical problem processes a user interface element so that a web page developer can focus on other aspects of a web page about property capsulation programming demanded.
[0007] However, server side processing of a client side user interface element may be typically accompanied by the complicated status management problem in a server side application program. The condition of the server side processing module corresponding to a client side user interface element expresses the property and configuration of a server side module. However, in the communication link between a client and a server, since the communication link connection between a client and a server is not expected but it may be cut, with a client / server model without status management, it will point so that a server may not maintain the condition of the process generated between a client and a server.
[0008]
[Means for Solving the Problem] According to this invention, **** and other technical problems are solved by offering the status management by which server side processing was encapsulated about the client side user interface element with which the server does not hold the condition between two requests. This is related with the condition (for example, "view state" (viewstate)) of one or more server side control objects of a control object hierarchy by processing of a client side user interface element including status management processing. A server side control object processes and generates the client side user interface element displayed on a web page. The hierarchy of a server side control object collaborates and generates an authoring language code a result like [again] the criterion HTML displayed on the web page of a client. As long as a client supports for example, the criterion HTML or another authoring language, it may be what kind of browser.
[0009] The status information of a server side control object has between a client and a server conveyed between two HTTP requests with the condition structure which can be conveyed in the client / server model which does not perform status management rather than is held at a server side (in order to fill). If a server receives a HTTP request from a client, a server will extract status information from the condition structure from a HTTP request which can be conveyed, and will distribute this status information to each suitable control object of a server side hierarchy. The integrity of the condition

structure which can be conveyed is also inspected in relation to the condition structure which can be conveyed using the coded structure which was generated from status information.
[0010] The status management approach of the server side control object corresponding to the client side user interface element built into the web page displayed on a client is offered. A server side control object is created to a control object hierarchy, and a client side user interface element is processed. The condition structure which can be conveyed is received from a client. The condition structure which can be conveyed includes the status information which shows the status value of a control object hierarchy's server side [at least one] control object. Status information is extracted from the condition structure which can be conveyed. When the status value from status information relates to the server side control object, the status value is loaded to the property of a server side control object.
[0011] The computer program product which manages the condition of two or more server side control objects corresponding to two or more client side user interface elements built into the web page which is created by the control object hierarchy and displayed on a client is offered. The condition structure which was received from the client and which can be conveyed includes the status information relevant to one or more server side control objects of a control object hierarchy. Status information is serialized and the hierarchy of the property of status value, the related property data type, and a server side control object is extracted. Based on a hierarchy, the server side control object in a control object hierarchy is discovered. Status value is loaded to the property of a server side control object.
[0012] The product as a computer program product is offered. According to a computer system, the operation gestalt with the computer program product by this invention can be read, and contains the computer program storage which codes the computer program which performs the computer process which manages the condition of the server side control object corresponding to the client side user interface element built into the web page displayed on a client. Shape is taken by the computer system by the subcarrier and another operation gestalt of the computer program product by this invention includes the computer data signal which codes the computer program which manages the condition of the server side control object corresponding to the client side user interface element built into the web page displayed on a client. The product generated by the process of this invention is offered as an authoring language code which is sent to a client and includes the condition structure which can be interpreted by the browser on a client, and which can be conveyed.
[0013]
[Embodiment of the Invention] In an operation gestalt with this invention, on a web server, web page contents are generated dynamically and displayed by the client. A client and a web server communicate by the network using for example, a HTTP request and a HTTP response. Thus, a web server generates web page contents, such as the HTML code gestalt, and transmits the contents to the client which can display a web page on a browser. Each user interface element of a web page and the server side control object which corresponds logically are created on a web server, and processes and generates the web page contents used by the client side browser which displays and processes a web page. A server side control object is declared in dynamic contents files, such as an ASP+ resource processed by the page factory which creates the hierarchy of a server side control object. The control object in a hierarchy collaborates and processes the request received from the client, subsequently, it generates result web page contents, transmits to a client, and terminates the control object in a hierarchy.
[0014] In an operation gestalt with this invention, a page object can be illustrated as a control object hierarchy's best level. In the page object which is also a control object, each child control object spreads to the hierarchy of a large number level including one or more child control objects of itself including one or more child control objects typically. A page object and a descendant's control object perform a processing sequence, and process and generate the web contents corresponding to a client side user interface element.
[0015] One of the processings in this sequence includes the load processing which un-serializes the status information received with the condition structure which is included in a request from a client, and which can be conveyed to suitable one or two or more suitable objects in a hierarchy. The traverse processing in this load processing traverses a control object hierarchy according to the hierarchy of the

condition structure which can be conveyed, in order to find a control object suitable in a hierarchy. in an operation gestalt with this invention, the received status information includes only the condition of the control object which changed from the initial state (namely, this status information -- difference -- only data are included). With another operation gestalt, if it will not be to change, there is a thing including some another combination of all the conditions of all the control objects in a hierarchy or one or more status information of a server side control object, and a hierarchy.

[0016] Another processing of the above-mentioned sequence includes the preservation processing which serializes the status information from the control object from which the condition changed. Status information adds status information to the condition structure which is transmitted to a client and which can be conveyed. A client returns the structure which can be conveyed to a web server at the next request to a web server. If the condition structure which can be conveyed is received, a control object hierarchy will be re-created and a hierarchy condition will return to the hierarchy condition of a former response by load processing.

[0017] It can avoid transmitting the condition structure which can be conveyed to a client in another operation gestalt. The condition structure which can be conveyed remains in a server, without being transmitted, and may be transmitted to resource somewhere else (for example, another server). For example, in the case of load balancing, the 1st server communicates with a browser by the 1st HTTP transaction, but the 2nd server can communicate with a browser by the 2nd HTTP transaction. Therefore, the condition structure which can be conveyed is sent to the 2nd server in which the 2nd carries out HTTP transaction processing from the 1st server, and can bypass a client by it.

[0018] Drawing 1 shows the web server which generates dynamically the web page contents displayed on the client in an operation gestalt with this invention. A client 100 performs the browser 102 which displays the web page 104 on the indicating equipment of a client 100. A client 100 may also include the client computer system which has a display like a video monitor. The "INTERNET EXPLORER" browser currently sold by Microsoft Corp. is an example of the browser 102 in an operation gestalt with this invention. As an example of other browsers, it is "NETSCAPE NAVIGATOR". It reaches. Although there is "MOSAIC" etc., it does not restrict to this. The text box control 106 and two carbon button control 108 and 110 are included in the illustrated web page 104. A browser 102 can receive the HTML code from a web server 116 by the HTTP response 112, and displays the web page described by the HTML code. Although HTML is explained with reference to a certain operation gestalt Not the thing restricted especially but SGML (Standard Generalized Markup Language), XML (eXtensible Markup Language) -- and It is the markup language of the XML base. It is thought that other authoring languages containing WML (Wireless Markup Language) designed so that the contents and the user interfaces of narrow-band radio equipment, such as a pocket bell (trademark) and a cellular phone, might be specified are within the limits of this invention. Furthermore, although the criterion HTML 3.2 is mainly indicated in this specification, any versions of HTML may be contained within the limits of this invention.

[0019] The communication link with a client 100 and a web server 116 can be performed using a series of processings of the HTTP request 114 and the HTTP response 112. Although HTTP is explained with reference to a certain operation gestalt, it is not restricted especially and it is thought that other transport protocols including S-HTTP are within the limits of this invention. In a web server 116, the HTTP pipeline module 118 receives the HTTP request 114, analyzes URL, and calls the suitable handler which processes a request. The web server 116 is equipped with two or more handlers 120 treating the resource of a different type in the operation gestalt with this invention.

[0020] For example, when URL specifies a static contents file 122 like an HTML file, a handler 120 accesses the static contents file 122, and sends the static contents file 122 to a client 100 by the HTTP response 112 through the HTTP pipeline 118. Or in an operation gestalt with this invention, when URL specifies a dynamic contents file 124 like an ASP+ resource, a handler 120 accesses the dynamic contents file 124, processes the contents of the dynamic contents file 124, and generates the HTML code the result for web page 104. In an operation gestalt with this invention, the result HTML code contains standard HTML3.2 code. Generally, a dynamic contents file is the server side declaration data storage

section (for example, ASP+ resource) which can use the authoring language which describes the web page which should be displayed on a client for generating dynamically. And the HTML code for web pages passes the HTTP pipeline 118, and is sent to a client 100 by the HTTP response 112.
[0021] During this processing, the handler 120 was beforehand developed again, in order to simplify a development effort, or it can access the library of the 3rd person code. One of such the libraries is the server side class control library 126, and a handler 120 can illustrate from here the server side control object which generates HTML data as a result of processing a user interface element and displaying on a web page. in an operation gestalt with this invention, one or more server side control objects are visible on the web page described by the dynamic contents file 124 -- it hides-like and maps to one or more user interface elements.
[0022] On the other hand, the 2nd library is a client side control class library 128 like the library containing the "ACTIVEX" component from Microsoft Corp. "ACTIVEX" control is a COM (component object model) object which follows a fixed criterion in the method of a dialogue with a client and other components. Client side "ACTIVEX" control is the component of the COM base as for which downloads to a client automatically and executive operation may be carried out to it by the web browser of a client. A server side ACTIVEX component (not shown) is a component of the COM base which may be performed on a server, in order to achieve the various server side functions in which the server side functionality of stock price retrieval application or a database component is offered. ACTIVEX is indicated by the detail by "understanding of ACTIVEX and OLE" (David Chappelle, the Microsoft press, 1996).
[0023] In contrast with "ACTIVEX" control, the server side control object of the operation gestalt of this invention specified as the dynamic contents resource 124 corresponds to the user interface element built into the web page on a client logically. A server side control object can generate the effective HTML code which may contain the locator which carries out the reference of the given client side "ACTIVEX" control to for example, a HTML tag again. When the browser has already had the code for client side "ACTIVEX" control in the storing system, executive operation of the "ACTIVEX" control is carried out within the web page on a client. If it becomes right [that], a browser will download the code for "ACTIVEX" control from the resource specified by the locator, and will carry out executive operation of the "ACTIVEX" control within the web page on a client. The server side control object in the operation gestalt of this invention can raise an event again to the server side "ACTIVEX" control used for performing stock price retrieval application on a server.
[0024] A handler 120 accesses one or more non-user interface server components 130 which carry out executive operation on a web server 116 or another accessible web server again. A non-user interface server component 130 like stock price retrieval application or a database component is referred to in the dynamic contents file 124 processed by the handler 120, or is related with it. The server side event raised by the control object declared by the dynamic contents file 124 may be processed in server side code which calls the suitable method of the non-user interface server component 130. Consequently, the processing offered by the server side control object can simplify programming of non-user interface server component stereo-NENTO 130 by encapsulating processing and generation of a web page of a user interface element, and, thereby, the developer of the non-user interface server component 130 can be concentrated on development of the function of the application proper instead of a user interface problem.
[0025] Drawing 2 shows the flow chart of processing of the client side user interface element using the server side control object in an operation gestalt with this invention, and generation processing. In processing 200, a client transmits a HTTP request to a server. A HTTP request contains URL which specifies resources, such as an ASP+ resource. In processing 202, a server calls the suitable handler which receives a HTTP request and processes the specified resource. A HTTP request includes the condition structure including a hierarchy which can be conveyed in one or more the status information and arbitration about a server side control object. However, the HTTP request of the beginning to a given page does not include typically the condition structure which can be conveyed from the condition not changing in the server of the given page. Furthermore, in case the condition structure which can be

conveyed checks the effectiveness of property data-type information and status information (that is, it is checked that status information has not been altered by the client), it may also contain the integrity code for assisting a server. Processing 204 generates a server side control object hierarchy based on the contents of the specified dynamic contents file (for example, ASP+ resource). An ASP+ resource is read in processing 203. In processing 205, the status information received with the condition structure which can be conveyed is loaded to the suitable server side control object in a hierarchy, and a control object is returned to a former condition.

[0026] In processing 206, a control object hierarchy's server side control object performs postback event handling, postback data handling, status management, and one or more processings in data coupling. From a client, the postback event and data (collecting "postback input") from a user interface element are sent to a server, and are processed. For example, especially a postback event is not restricted and may also contain the "data change" event from the client side text box element sent to "a mouse click" or a server from a client side carbon button element. Especially postback data are not restricted and may also contain the text inputted into the index of the item chosen from the text box element or the drop down box by the user.

[0027] In processing 209, a property value, related type information, and a hierarchy are stored in the condition structure which is transmitted to a client and which can be conveyed. In processing 208, in order that each server side control object in a hierarchy may generate authoring language data like the HTML code for the display by the web page of a client side user interface element (or rendering), it is called. Although the vocabulary "a rendering" may mean the processing which displays graphics on a user interface, in this specification, the "rendering" also means generation processing of the authoring language data which may be interpreted by client application like the browser for a display and a client side function. The rendering processing 208 generates the authoring language data showing the condition structure which can be conveyed again. More detailed explanation of processing 206 and the rendering processing 208 is given in connection with drawing 6 . In a certain operation gestalt, the call of the render() method in each control object is performed using a tree traversal sequence. That is, the call of the render() method of a page object becomes the repetitive traverse covering the suitable server side control object in a hierarchy. As an option which calls the render() method of a suitable control object, approaches, such as event signaling or the object registration technique, may be used. A parenthesis specifies the "render()" level which shows the method of comparing with a data value.

[0028] In an operation gestalt with this invention, actual creation of each server side control object may delay until a server side control object is accessed in processings 206 or 208 (handling of a postback input, loading of a condition, rendering of a control object to the HTML code, etc.). When a server side control object is not accessed for a given request, server processing is optimized by delaying creation of a control object and eliminating unnecessary control object creation processing.

[0029] In processing 210, authoring language data (for example, the HTML code) including the condition structure which can be conveyed are transmitted to a client by the HTTP response. In processing 214, a client receives the HTML code including the condition structure relevant to the web page which should be displayed which can be conveyed. The condition structure which can be conveyed may be saved at a client, in order to return to a server at the next HTTP request. In processing 216, a client system incorporates the user interface element of a new page according to the HTML code received from the HTTP response (for example, it displays). however, a user interface element -- incorporating -- he should also understand that you may include non-display processing of control of read-out to the memory which offers voice or a tactile-sense output and writing, and script processing etc. A server side control object hierarchy is ended in processing 212. In an operation gestalt with this invention, the server side control object in a hierarchy answers the HTTP request which refers to the associated ASP+ resource, is created, and after the rendering of authoring language data (for example, HTML data) finishes, it is destroyed. In another operation gestalt, processing 212 may be performed before processing 210 after processing 208.

[0030] Drawing 3 shows an example of the module in the web server used in an operation gestalt with this invention. A web server 300 accepts the HTTP request 302 in the HTTP pipeline 304. The HTTP

pipeline 304 may also contain various modules, such as logging of web page statistics, user collating, a right of user access, and a module for the formation of an output cache of a web page. Finally each input HTTP request 302 received by the web server 300 is processed by the specific instance of an IHTTP handler class (it illustrates as a handler 306). A prefix called IHTTP shows that the class is the interface of a HTTP handler. A handler 306 analyzes a URL request and calls a suitable handler factory (for example, page factory module 308).

[0031] In drawing 3 , the page factory 308 related with the ASP+ resource 310 is called, and the instantiation and configuration of an object which were declared in the ASP+ resource 310 are handled. In a certain operation gestalt, an ASP+ resource may be recognized or referred to by specifying a specific suffix (or a file extension part like ".aspx") as a file. If the request to given ".aspx" resource is first received by the page factory module 308, the page factory module 308 will search a file system, and will obtain a suitable file (for example, .aspx file 310). This file may also contain the text (for example, authoring language data) which may be behind interpreted or accessed by the server which processes a request, or the data (for example, cutting tool code data or coded data) in another format. When a physical file exists, the page factory module 308 opens a file and reads the file to memory. When a file is not found, the page factory module 308 returns the suitable error message "a file is not found."

[0032] After reading the ASP+ resource 310 to memory, the page factory module 308 processes a file content, and builds the data models (for example, a list [of script blocks], directive, static text area, and hierarchy server side control object, a server side control property, etc.) of a page. A data model is used for generating the source list of a new object class like the COM+ (Component Object Model+) class to which the class of the page base is made to extend. The page base class contains the code which specifies the structure, property, and function of a basic page object. Subsequently, a source list is dynamically compiled by intermediate language. Intermediate language is COM+ IL. A code and Java A cutting tool code and Modula 3 A code and SmallTalk A general purpose or custom-made orientation linguistic code of a code, the Visual Basic code, etc. may also be included. In another operation gestalt, intermediate-language processing may be excluded and the instruction of a proper may be directly generated from a source list or a source file (for example, ASP+ resource 310). The control class library 312 may be accessed with the page factory module 308, in order to obtain the server side control class which is used for a control object hierarchy's generation and which was defined beforehand.

[0033] The page factory module 308 outputs the page object 314 which is a server side control object equivalent to the web page 104 of drawing 1 . The page object 314 and its child object (for example, the text box object 318, the carbon button object 320, and another carbon button object 322) are the control object hierarchy's 316 examples. The example of other control objects can be considered according to this invention, is not restricted especially, and contains the object and custom-made control object corresponding to HTML control of Table 1. The page object 314 corresponds to the web page 104 of drawing 1 logically. The text box object 318 corresponds to the text box 106 of drawing 1 . Similarly, the carbon button object 320 corresponds to the additional carbon button 108 of drawing 1 , and the carbon button object 322 corresponds to the deletion carbon button 110 of drawing 1 . The page object 314 relates to other control objects and hierarchy targets on a server. In a certain operation gestalt, a page object is a container object which contains the child control object hierarchical. The hierarchical relationship of other gestalten, such as a dependency, can be used with another operation gestalt. In a complicated control object hierarchy, one child object may be a container object of other child objects rather than it has the child object of a large number level.

[0034] It sets in the above-mentioned operation gestalt, the control object hierarchy's 316 control object is created and performed on a server 300, and each server side control object "carries out the mirror of the user interface element with which it corresponds on a client." The server side control object of this operation gestalt collaborates, handles the input from the HTTP request 302 again, manages the condition of a server side control object, performs data coupling with a server side database, and generates the authoring language data (for example, the HTML code) used for the display of a web page as a result of a client. Result authoring language data are generated from the server side control object hierarchy 316 (namely, rendering), and are transmitted to a client by the HTTP response 324. For

example, the result HTML code can embody any effective HTML configurations, and in addition to this, control of an ACTIVEX type, a JAVA (trademark) applet, a script, and when it is processed by the browser, refer to any web resources which produce client side user interface elements (for example, a control carbon button, a text box, etc.) for it.
[0035] By declaration made with the ASP+ resource 310, a server side control object can access one or more non-user interface server components 330 for a dialogue with the non-user interface server component 330 and a client side user interface element. For example, a postback input can be answered and a server side control object can raise a server side event to the non-user interface server component registered into those events. Thus, the non-user interface server component 330 minds a user interface element for a dialogue with a user, and it can perform it, without programming a code required displaying and processing these elements.
[0036] In short, the operation gestalt of this invention contains on a server creation and the server side control object by which executive operation is carried out, in order to generate the HTML code sent to a client. The HTML code can embody any effective HTML configurations, and is control of an ACTIVEX type and JAVA, for example. Any web resources of the others which generate the user interface carbon button and other user interface elements in an applet, a script, and a client can be referred to. The user in a client can have a dialog with these user interface elements that correspond to a server side control object logically, and can return a request to a server. A server side control object is re-created on a server, and it processes the data, the event, and other properties of a user interface element so that the HTML code of the next round which should transmit to a client as a response may be generated.
[0037] If drawing 4 is referred to, an example of the computer system of the operation gestalt of this invention contains the general purpose computer equipment of the gestalt of the conventional computer system 400 containing the system bus 406 which connects the various system components containing the processor unit 402, a system memory 404, and a system memory 404 to the processor unit 400. System buses 406 may be any of the bus structure of some types containing the peripheral bus and local bus which use a memory bus or a memory controller, and various bus architectures. The system memory contains the memory (ROM) 408 only for playbacks, and random access memory (RAM) 410. The unformatted input / output system 412 (BIOS) containing the basic routine which helps a transfer of the information between the elements within a computer system 400 are stored in ROM408.
[0038] The computer system 400 contains the optical disk drive 418 which performs read-out and the writing of a removable optical disk 419 like the magnetic disk drive 414 and CD ROM which perform further read-out and the writing of a hard disk drive 412 and the removable magnetic disk 416 which perform read-out and the writing of a hard disk, DVD, or other optical media. The hard disk drive 412, the magnetic disk drive 414, and the optical disk drive 418 are connected by the hard disk drive interface 420, the magnetic disk drive interface 422, and the optical disk drive interface 424 system bus 406, respectively. A drive and its medium which can be related computer read offer the instruction of a computer system 400 which can be computer read, DS, a program, and the non-volatile storage section of other data.
[0039] Although the hard disk, the removable magnetic disk 416, and the removable optical disk 419 are used for this specification in the above-mentioned environmental example of a publication, the medium type [other] in which data storage is possible and which can be computer read can be used for the above-mentioned example of a system. The medium of other types of these which can be used for the above-mentioned example of operating environment which can be computer read has for example, a magnetic cassette, flash memory card, a digital videodisc, a BERUNUI (Bernoulli) cartridge, random access memory (RAM), the memory (ROM) only for playbacks, etc.
[0040] Many program modules may be stored in a hard disk, a magnetic disk 416, an optical disk 419, and ROM408 or RAM410, and these contain the application program 428, other program modules 430, and the program data 432 of 426 or 1 or more operating systems. A user can input a command and information into a computer system 400 with input units, such as a keyboard 434 and a mouse 436, or other pointing equipments. As other input devices, there are a microphone, a joystick, a gamepad, a

satellite dish, a scanner, etc., for example. These and other input devices are connected to the processor 402 through the serial port interface 440 connected to the system bus 406 in many cases. However, these input devices may be connected by other interfaces, such as a parallel port, a game port, or Universal Serial Bus (USB), again. The monitor 442 or the indicating equipment of other types is also connected with the system bus 406 through the interface of a video adapter 444 etc. In addition to a monitor 442, typically, a computer system contains other circumference output units (not shown), such as a loudspeaker and a printer.

[0041] A computer system 400 can operate in the environment using the logical connection to one or more remote computers like a remote computer 446 connected by network. a remote computer 446 -- a computer system, a server, a router, Network PC, and a pier (peer) -- it may be equipment or other common network nodes, and there are many elements typically mentioned above in connection with a computer system 400, or all are included. Network connection contains Local Area Network (LAN) 448 and Wide Area Network (WAN) 450. Such a network environment is not new in office, a company scale computer network, intranet, and the Internet.

[0042] When using by the LAN network environment, a computer system 400 is connected to a local network 448 through a network interface or an adapter 452. When using by the WAN network environment, a computer system 400 includes the modem 454 or other means for establishing the communication link by Wide Area Network 450 like the Internet typically. Built-in or external any is sufficient as a modem 454, and it is connected with the system bus 406 through the serial port interface 440. In the environment connected by network, the program module described in relation to the computer system 400 or its part may be memorized by remote memory storage. The illustrated network connection is an example and can use other means for the communication link establishment between computers.

[0043] In the operation gestalt of this invention, a computer 400 expresses a web server and CPU402 carries out executive operation of the page factory module on the ASP+ resource memorized by at least one of storages 416, 412, 414, 418, and 419 or memory 404. A HTTP response and a request communicate by LAN448 connected to the client computer 446.

[0044] Drawing 5 is a process flowchart showing server side processing of the page object in an operation gestalt with this invention, and other control objects. In processing 500, the page object construction section is called with the page factory module 308 (refer to drawing 3). Consequently, it is created in order that a page object (see the page object 314 in drawing 3) "may carry out the mirror of the web page user interface element on a client." In processing 502, a page factory module calls the ProcessRequest member function of the page object which begins to process [gradual] the HTTP request received from the client. In the 1st phase of 1 operation gestalt of this invention, server side creation processing (not shown) may be creation of the descendant server side control object contained in the control object hierarchy of a page object. That is, in order that the construction section of a child control object may create the control object between the processing life times of HTTP request processing, call appearance is carried out repeatedly.

[0045] However, in another operation gestalt, creation of a child control object is delayable until a control object is needed for a given processing step (for example, rendering of the HTML code of the user interface element with which handling of a postback event, handling of postback data, loading of a view state and preservation, and data coupling analyze or correspond). Since the operation gestalt of the latter "which delays control object creation" can reduce use of unnecessary CPU and memory, it is the optimal. For example, there is a case where it will be called creation of the web page from which the user input event received from the client completely differs. In this case, a control object hierarchy makes it end immediately, and only in order to process the event which will illustrate a new different control object hierarchy of a new page, it is not necessary to illustrate the whole control object hierarchy of a former page.

[0046] The server call of the ProcessRequest method of a page object can be answered, and executive operation of the processings 504-520 can be carried out by the data page object and each descendant control object according to the data of a given HTTP request etc. In an operation gestalt with this

invention, processings 504-520 are performed to each object of each in the sequence of drawing 5 . However, depending on a HTTP request, about given processing of another object, it may not be carried out in order or, as for the given processing to one object, processing may not be performed at all. For example, the 1st object performs the initialization processing 504 and its load processing 506, the postback data processing 508 is started and a descendant control object performs the initialization processing 504 and the load processing 506 of itself by overdue control object creation after that. Although the sequence of processing by the page object and the descendant control object is not restricted to this, it depends on the various factors [whether the current condition of the property of the data in a HTTP request, a control object hierarchy's configuration, and a control object and control object creation are performed behind time] to include.

[0047] The initialization processing 504 initializes a control object, after a control object is created by carrying out executive operation of the server side code of the arbitration about initialization in a dynamic contents file. Thus, each server side control object may be customized by the specific server side function declared by the dynamic contents file. In the operation gestalt of this invention, it was meant that a dynamic contents code customizes or extends the base page control class declared by the page developer with the ASP+ resource on a server. Compile of an ASP+ resource contains the declared code in the suitable initial code (for example, initialization () method of a page object and a descendant control object). The initialization processing 504 carries out executive operation of this code, and customizes or extends a page base class and the base class of a descendant control object.

[0048] In an operation gestalt with this invention, the status management of a server side control object is supported by the load processing 506 and the preservation processing 516 in which the condition structure which can be conveyed is used, in order to contain the non-statement model of a client/server system by returning a server side control object to a former condition. With a certain operation gestalt, although a condition goes and comes back to a server in the one or more HTML fields in which the HTTP request / response of a pair hid, it is considered that other condition structures which can be conveyed, such as Cookie and the visible field, are within the limits of this invention.

[0049] In a series of processings by the given present request and present given response about a page between a client and a server, the condition of one or more control objects is recorded on the condition structure which can be conveyed by the preservation processing 516 after processing of a previous request. In an operation gestalt with this invention, the status information of the addition containing the control object identifier which enables a hierarchy or a server to associate a suitable control object and a given condition is also included in the condition structure which can be conveyed. In the next HTTP request, status information is returned to a server with the condition structure which can be conveyed. A server extracts status information from the received condition structure which can be conveyed, loads condition data to the suitable control object in a control object hierarchy, and returns each control object to the condition that it existed before the previous HTTP response. After the processing to the present request, again, the condition of one or more server side control objects is recorded on the condition structure which can be conveyed by the preservation processing 516, and returns the condition structure which can be conveyed to a client by the next HTTP response by it.

[0050] Each server side control object is changed into the same condition as the condition before a previous HTTP request as a result of the load processing 506. For example, when a text box control object includes a property value equivalent to "JDoe" before a previous HTTP response, the load processing 506 returns the same control object to a previous condition by loading text string "JDoe" to the property value etc. Furthermore, whether the condition of a given object was memorized and recovered can also constitute.

[0051] When 1 operation gestalt of this invention is summarized, "it is saved" after the condition of one or more server side control objects processing. The saved status information is transmitted to a client by the response. A client returns the status information saved by the next response to a server side. A server loads status information to the newly illustrated server side control object hierarchy so that a hierarchy's condition may return to a former condition.

[0052] With another operation gestalt, status information can be held for some other accessible web

locations by the server during round trip on a server of returning from a server to a client and a server. After a server receives a client request, this status information is taken out by the server and may be loaded to the suitable server side control object in a control object hierarchy.
[0053] In processing 508, the postback data received from the HTTP request are processed. Postback data may be contained in the payload of the HTTP request by the key value and hierarchical display (for example, XML) which became a pair, or other data display like RDF (Resource Description Framework). Processing 508 analyzes the syntax of a payload and identifies the identifier of the meaning of a server side control object. When an identifier (for example, "page1:text1") is found and the identified server side control object exists in a control object hierarchy, corresponding postback data are passed to the control object. For example, reference of drawing 1 sends the identifier of a meaning relevant to a text box 106 and text "JDoe" to a web server 116 with the payload of the HTTP request 114. Processing 508 analyzes the syntax of the payload of the HTTP request 114, and acquires the identifier and its related value (namely, "JDoe") of a meaning of a text box 106. And processing 508 analyzes the identifier of the meaning of a text box 106, identifies a corresponding server side control object, and passes a "JDoe" value to the object to process.
[0054] As the load processing 506 was explained, the property value of a server side control object may be returned to a former condition. If postback data are received, a server side control object will judge whether the property value of the point to which the passed postback value corresponds was changed. When it changes, logging of the change is carried out to the change list in which data change of a related control object is shown. After all postback data are processed within a control object hierarchy, a control object method is called and one or more postdata change events can be raised to one or more non-user interface server components like stock price retrieval application started on a server. A postdata change event is an event which shows that for example, postback data changed the property of a server side control object. In an instantiation operation gestalt, such an event is sent to a system offer event queue, and can call the server component registered so that an event might be processed. Thus, a server side non-user interface server component can answer the event in which the trigger was carried out by change of the data of a server side control object. The option including using an application offer event queue, polling, and processing interruption which performs an event can also be considered within the limits of this invention.
[0055] A postback event is handled in processing 510. A postback event communicates with the payload of a HTTP request. Processing 510 analyzes the syntax of the specific event target (for example, with the operation gestalt with this invention, label attachment is carried out with "##EVENTTARGET") which identifies the server side control object to which the event is turned. Furthermore, processing 510 analyzes the syntax of it, when there is a discovered event argument, and it gives it to the server side control object which had the event argument (for example, label attachment is carried out with "##EVENTARGUMENT" with the operation gestalt with this invention) specified. A control object starts the event processed with the non-user interface server component (for example, server side stock price retrieval application) relevant to a dynamic contents file.
[0056] Processing 512 solves data coupling between a server side control object and one or more databases with an accessible server. the property of a server side control object may be related with the property of a parent data-coupling container as shown in the table of a server side application database with an operation gestalt with this invention (or data coupling is carried out -- having). A page frame work piece can update the control object property which has the value of the parent data-coupling container property corresponding to between the data-coupling processings 612 and by which data coupling was carried out. Thus, the user interface element on the web page in the next response reflects the updated property value correctly. The control object property with which a user interface element corresponds is because it was automatically updated between the data-coupling processings 512. Similarly, a control object property may be updated by the parent data-coupling container field, and, thereby, updates a server side application database in the postback input from a server side control object again.
[0057] Processing 514 updates a large number by which executive operation may be carried out, before

saving a control object condition and carrying out the rendering of the output. Processing 516 requires status information (namely, view state) from one or more control objects in a control object hierarchy, stores status information, and inserts it in the condition structure which is sent to a client with a HTTP response payload and which can be conveyed. For example, a "grid" control object can save the present index page of the list of values, and a "grid" control object can return to this condition after the next HTTP request (namely, processing 506). As mentioned above, view state information expresses the condition of the control object hierarchy before the next action by the client (for example, before sending a HTTP response to a client). It will be used for changing a control object hierarchy into the condition of the point before client post back input process or data coupling if view state information returns.

[0058] The rendering processing 518 generates the suitable authoring language output (for example, HTML data) sent to a client by the HTTP response. A rendering is performed by top-down hierarchy tree WOKU and the embedded rendering code of all server side control objects. Processing 520 does the last clean-up activity (it is connection of a database in closing a file ****) of arbitration, and a control object hierarchy is terminated. Subsequently, processing performs return and processing 522 to 502, and a page object is ended by calling the destructive section there.

[0059] Drawing 6 shows an example of the part of the dynamic contents file (for example, ASP+ resource) in an operation gestalt with this invention. Rhine 1 of the ASP+ resource 600 is the initiation tag of an HTML file, and is declared as a literal in the ASP+ resource 600. A literal corresponds to the server side literal control object in a server side control object hierarchy. Since it is the 1st control object by declaration with the ASP+ resource 600, an index "0" is given to the literal control object corresponding to Rhine 1. In rendering time amount, a literal control object only generates new Rhine for only containing in a "<html>" text and a HTTP response. Rhine 2-12 of the ASP+ resource 600 expresses a code declaration block, and executive operation of this is carried out on a server (namely, "runat =server" of Rhine 2 it was shown by the attribute like). In the example of this ASP+ resource 600, instantiation of the server side control object in a control object hierarchy does not take place as a result of a code declaration block. That is not right, and by code declaration block, it "be connected with a wire" connect [with the server side control object declared in Rhine 15] Is sufficient for a server side code, or it is associated. Rhine 13 of the ASP+ resource 600 is the initiation tag of the text of an HTML file, is declared with the ASP+ resource 600 as a literal, and serves as a literal control object which has an index "1."

[0060] Declaration in Rhine 14 declares the server side form control object (index = "2") illustrated by the control object hierarchy. The tag in Rhine 17 and 18 is a termination tag expressed as the server side literal control object (index = "3") within a control object hierarchy. The termination tag of Rhine 16 ends the <form> declaration corresponding to a form control object.

[0061] On the following hierarchy level, declaration of Rhine 15 declares the server side label control object (index = "1") which has the identifier attribute ("id") of "Message" using HTML control tag "span". The index "0" of this hierarchy level and "2" are assigned to the literal control object (index = "2") of the white space literal a literal control object (index = "0") and next to the white space literals (for example, a tab, new Rhine, a tooth space, etc.) of precedence, respectively.

[0062] Drawing 7 shows a code, as a result of answering the partial example of the dynamic contents file of drawing 6 and being generated by one or more server side control objects. In order to simplify this explanation, only form, status management, and a span section are shown in the HTML code section 700. Although other tags like <html> and <body> are not contained in the HTML code section 700, the HTML code of these tags may be contained in 1 set of perfect HTML codes from the ASP+ resource of drawing 6 .

[0063] Rhine 1 and 5 is initiation and the termination tags of the form section of the HTML code, and is generated by the server side form control object declared in Rhine 14-16 of drawing 6 . Rhine 4 contains the HTML code of the label (namely, "Last Post Back:5/6/99") which shows the date of the last post back processing declared in Rhine 15 of drawing 6 .

[0064] The field in which Rhine 2 and 3 was hidden expresses 1 operation gestalt of the condition

structure by this invention which can be conveyed. In Rhine 2, the status information of the server side control object declared with the ASP+ resource 600 is recorded on the field in which the name of "##VIEWSTATE" hid. A "##VIEWSTATE" field value is a text string showing the hierarchy of the control object in status value, a property type, and a control object hierarchy. Since load processing traverses a hierarchy and loads given status value to the property of the suitable control object in a hierarchy again, the hierarchy is included in the "##VIEWSTATE" field.
[0065] With an operation gestalt with this invention, a client has a dialog with a web server using a HTTP request / response pair sequence. Between the response to a client, and the next request which a server receives, a server does not need to maintain the condition of the server side control object relevant to connection with a given ASP+ resource or a given given client. Instead of maintaining, it is returned to a client with the condition structure which can be conveyed (by for example, HTTP response), and a given control object hierarchy's status information is returned by the server with delivery and the condition structure which can be conveyed (at for example, the next HTTP request).
[0066] Rhine 3 expresses the file in which the name of "##VIEWSTATE" which has the value "434333433" hid. "##VIEWSTATEMAC" is an integrity code in the condition structure which can be conveyed where it is used by load processing, in order to check that the ##VIEWSTATE value is not altered by the client. # At first, #VIEWSTATEMAC value is calculated by the server from the contents of the ##VIEWSTATE value, and is sent to a client by the response. When a client returns the condition structure which can be conveyed, a web server calculates a new integrity code from the contents of the received ##VIEWSTATE value. if the received integrity code and the newly calculated code are equal, the ##VIEWSTATE value which the web server received is effective -- or it is not altered -- ** (that is, the same as the ##VIEWSTATE value previously sent to the client) -- it judges. With an operation gestalt with this invention, an integrity code is calculated using an MD5 algorithm. MD5 is the algorithm created by Professor Donald RIBESUTO in 1991, in order to use it for creation of a digital signature. The hash function of an one direction, i.e., a message, is changed into reception, it changes it into a digital fixed string, and MD5 is also called a message digest. Such a method may also include use of the secret key which reduces the unjust actuation by an individual or a program without an access privilege. However, with other operation gestalten, you may also include another integrity coding technique containing the die length of generating the code based on MD4 algorithm or the hash algorithm of other arbitration, or the ##VIEWSTATE field.
[0067] The structure of the ##VIEWSTATE field in Rhine 2 expresses a control object and the hierarchical nest condition of those conditions (for example, status value and the property data type relevant to it) with an operation gestalt with this invention. # The tag used for an example of #VIEWSTATE field is shown in (Table 1). The tag containing "/" expresses a termination tag.
[0068]
[Table 1]

タグ	説明
<s>, </s>	ストリング値
<a.x>, 	x 要素を有するアレイ
<i>	整数値
<h.x>, </H>	x テーブルエントリを有するハッシュテーブル
<K>, </K>	ハッシュテーブルキー
<V>, </V>	ハッシュテーブル値
,	ブーリアン値
<d>, </d>	日付／時間値
<c>, </c>	カレンシ値
<A>, </A>	アレイリスト
<n>, </n>	ヌル値

[0069] Although an example of a coding implementation gestalt is shown in this specification, with another operation gestalt, he should understand that other coding approaches can be used. For example, a hierarchy, status value, and those related property data types can also be specified using the hierarchy identifier of the meaning of each server side control object, or the data of the condition structure which can be conveyed can also be expressed using an XML related data format. Furthermore, using well-known encryption and well-known compression technology, safety can be secured and the size of the condition structure which can be conveyed can also be reduced.
[0070] Drawing 8 shows the control object hierarchy corresponding to the ASP+ resource of drawing 6 in an operation gestalt with this invention, and the ##VIEWSTATE field value of drawing 7 . Although it is not a control object hierarchy's control object component, it is shown as the best level of the hierarchy based on the ##VIEWSTATE field structure where the file check block 802 is shown in drawing 7 . More detailed explanation of the file check block 802 is given about drawing 9 . Or the file check block 802 is excluded or it is also thought that it is within the limits of this invention to make it together with the hierarchy level corresponding to the page control object 804.
[0071] The page control object 804 is illustrated as a control object hierarchy's best level, and corresponds to the special control object corresponding to the web page of a result itself. With an operation gestalt with this invention, especially the declaration (namely, declaration which has a "runat = server" attribute) for making the page object 804 illustrate is not needed.
[0072] In the following hierarchy level, the control objects 806, 808, 810, and 812 are contained in the page object 804, or are illustrated as a child control object which has relation hierarchical. The index of a zero base is given to each control object of one hierarchy level based on the sequence to the bottom [from / when it can set to an ASP+ resource]. for example, -- a literal -- control -- an object -- a hierarchy -- 806 -- drawing 6 -- ASP -- + -- a resource -- 600 -- Rhine -- one -- it can set -- " -- < -- html -- > -- " -- a text -- and -- accompanying -- white space -- corresponding -- **** . Therefore, the index "0" is given to the literal control object 806. With the above-mentioned operation gestalt of this invention, the code declaration block extended from Rhine 2-12 does not become instantiation of a corresponding control object. a literal -- control -- an object -- 808 -- drawing 6 -- ASP -- + -- a resource -- 600 -- Rhine -- 13 -- it can set -- " -- < -- body -- > -- " -- a text -- and -- accompanying -- white space -- corresponding -- **** . Therefore, the index "1" is given to the literal control object 808. The form control object 810 supports the form declaration in Rhine 14-16 of the ASP+ resource 600 of drawing 6 . Therefore, the index "2" is given to the form control object 810. The literal control object 812 supports the termination tag in Rhine 17 and 18 of the ASP+ resource 600 of drawing 6 , and attendant white space. Therefore, the index "3" is given to the literal control object 812.
[0073] In the following hierarchy level, the control objects 814, 816, and 818 are contained in the form

object 810, or are illustrated as a child control object which has relation hierarchical. The literal control object hierarchy 814 supports the white space before the span declaration in Rhine 15 of the ASP+ resource 600 of drawing 6 . Therefore, since the literal control object 814 is the 1st literal text (it is white space) which encounters in relation with this hierarchy level in an ASP+ resource, the index "0" is given to it. Similarly, the literal control object 818 supports the white space following the span declaration in Rhine 15 of the ASP+ resource 600 of drawing 6 . Therefore, the index "2" is given to the literal control object 818. The label control object 816 supports the declaration in Rhine 15 of the ASP+ resource 600 of drawing 6 . Therefore, the index "1" is given to the label control object 816.

[0074] Drawing 9 shows the nest version of the ##VIEWSTATE field value of drawing 7 . The nest version is indicated in order to simplify explanation of the hierarchical property of a ##VIEWSTATE field value into this specification. Rhine 1 of the nest version 900 corresponds to the beginning of the file check block 802 which became the nest of drawing 8 . Rhine 1 specifies 2 element array and contains 2 element array of Rhine 3 as the string and the 2nd element of Rhine 2 as the 1st element. Rhine 2 specifies the string value showing a file check code. With an operation gestalt with this invention, a file check code is calculated as a hash value of the ASP+ resource (for example, ASP+ resource of drawing 6) of relation. The array which starts in Rhine 3 expresses the beginning 804 of the status information of the following hierarchy level, i.e., the page control object of drawing 8 . The zero value of Rhine 4 expresses the 1st element of 2 element array of Rhine 3 which can hold the status information about the property of page object itself. In the above-mentioned operation gestalt, since the status information of a page object is not recorded, the value is "null" (zero).

[0075] As shown in Rhine 5, the 2nd element of 2 element array of Rhine 3 shows the hash table including the status information of the child (namely, control objects 806, 808, 810, and 812 of drawing 8) of a page control object. The hash table of Rhine 5 contains one hash table entry indicated to be "1" to a tag <H1>. <K> of Rhine 6 and 8 and a </K> tag contain the integral value "2" shown in Rhine 7 equivalent to the key of a hash table entry, i.e., the index of the form control object 810 of drawing 8 . The ##VIEWSTATE value shown in drawing 9 does not include the status information of the literal control objects 806, 808, and 812 of drawing 8 . Or when other control objects or child control objects which saved status information exist in the hierarchy level same as a form control object 810, the hash table of this level may have one or more hash table entries for containing the control object of an addition of this level, and the key to each additional hash table entry may be the corresponding index of a control object.

[0076] A hash table entry value is included in <V> of Rhine 9 and 24, and a </V> tag, and contains another 2 element array which specifies the condition of a form control object and its child object. The 1st element of the array specifies the status information where the form control object (that is, expressed with the zero value in this operation gestalt) was saved.

[0077] The 2nd element specifies another 1 entry hash table showing the status information of the child object of a form control object. <K> of Rhine 13 and 15 and a </K> tag contain the integral value "1" shown in Rhine 14 equivalent to the key of one hash table entry, i.e., the index of the label control object 816 of drawing 8 . The ##VIEWSTATE value shown in drawing 9 does not include the status information of the literal control objects 814 and 818 of drawing 8 .

[0078] A hash table entry value is included in <V> of Rhine 16 and 21, and a </V> tag, and contains another 2 element array which specifies the condition of a label control object and its child object. The 1st element of the array specifies the status information where the property of the label control object whose string value is "InnerHtml=5/16/99" specified in Rhine 18 was saved. A property name and "InnerHtml" are equal to status value"5/16/99", and are a "string" type. A semicolon expresses termination of the status value of a given property. As for each condition data value and its related property, a boundary is defined by the termination semicolon when the label control object of Rhine 18 contains the additional property with which the condition was saved. The zero value shown in Rhine 19 like the 2nd element of an array shows that the label control object does not contain the child object. Rhine 20-27 is the termination tags of the initiation tag which drawing 9 precedes.

[0079] With another operation gestalt, the serialization format called a finite object serialization

(Limitted Object Serialization (LOS)) format is used for the condition structure which can be conveyed. Generally, a LOS format specifies the hash table of an identifier / value pair of each control object in which each hash entry contains either of the hash tables of the status information of the property of a control object, and the nest of a child control object.
[0080] Table 2 shows an example of the syntax of a LOS format.
[0081]
[Table 2]

制御オブジェクト	• value type-table$_{opt}$ name-table$_{opt}$	• h<name1;value1>¥t50System.Drawing.Color¥n1BackColor
値	• typed-value • untyped-value • typed-array-value • untyped-array-value • untyped-hashtable-value	• 50<red> • <red> • a50<red;blue;green> • a<red;blue;green> • h<name1;value1>
タイプされた値	• type-ref value-list-start value-ref value-list-end	• 50<red>
タイプされたアレイ値	• array-modifier type-ref value-list-start array-value-ref value-list-end	• a50<red;blue;green>
非タイプ値	• value-ref	• <red>
非タイプアレイ値	• array-modifier value-list-start array-value-ref value-list-end	• a<red;blue;green>
値リスト開始	• <	
値リスト終了	• >	
値リストセパレータ	• ;	
アレイモディファイア	• a	
ハッシュテーブルモディファイア	• h	
非タイプハッシュテーブル値	• hashtable-modifier value-list-start hashtable-value-ref value-list-end	• h<name1;value1>
値の参照	• string-value • bin-ref base64-persisted-object	• ¥"This is a string value.¥"
バイナリ参照	• escape-char b	• ¥b
アレイ値参照	• value$_1$;[value$_2$;[value$_n$]]	• red;blue;green
ハッシュテーブル値の参照	• name-ref$_1$;value$_1$ [value-list-separator name-ref$_2$;value$_2$ [value-list-separator name-ref$_n$;value$_n$]]	• <name1;value1; name2;value2; name3;value3>
名前の参照	• string-name-number • string-name	• 1 • BackColor
タイプの参照	• known-type-number • string-type-number • string-type	• 10 • 50 • System.Drawing.Color

[0082] The following entries are considered as the 1st example of the operation of a LOS format. h<name1;value1;name2;value with \; escaped \> \"characters\"> This specifies the hash table of two pairs of an identifier/values, and all values here are strings. The 1st value is "name1". It is the name to say and is equal to string"value1." The 2nd value is a name of "name2", and is string"value with. ; It is equal to escaped > "characters"."
[0083] The following entries are considered as the 2nd example of the operation of a LOS format. h<control1;h<1;50<blue>;text;hello>control -- 2;h<1;50<red>;control3;> \t50System.Drawing.Color\n -- this specifies the hash table which has the three hash entries "control1", "control2", and "control3" 1

BackColor. The 1st hash entry is a name of "control1", and contains the child hash table which has two entries which are an identifier / value pair. The index "1" for referring to the name of "BackColor" specified at the last of this example is used for the 1st entry name of control1 hash table. The 1st entry value is a "System.Drawing.Color" type which is specified by the index "50" equally to "blue", and is specified near the last of this example. The identifier / value pair of the 2nd entry of control1 hash table contain identifier "text" and value "hello". The hash entry of "control2" contains the child hash table which has one hash entry which uses the index "1" for referring to the name of "BackColor" also here. This value is equal to "red" and is a "System.Drawing.Color" type. The hash entry of "control3" is empty. Although a hierarchy is specified by the nest of a hash table by the same approach as an above-mentioned operation gestalt, it is thought that the description approach of other hierarchies including use of a hierarchy identifier is also within the limits of this invention.

[0084] Drawing 10 shows the process flowchart which loads the status information memorized reception and there in the condition structure in an operation gestalt with this invention which can be conveyed to a control object hierarchy's control object. With the illustrated operation gestalt, unless at least one property of a server side [at least one] control object is changing, the condition structure which can be conveyed is not transmitted from a server (therefore, not returned to a server by the client). Since it is such, it is assumed that the illustrated flow chart exists in the condition structure which at least one element of status information received and which can be conveyed. With another operation gestalt, even if there is no status information in the condition structure which can be conveyed, some which go back and forth between a server and clients will have the condition structure which can be conveyed. With such an operation gestalt, when there is no status information in the condition structure which can be conveyed, decision processing (not shown) can operate so that a status management process may be closed.

[0085] Reception 1000 receives from a client the condition structure which can be conveyed at a HTTP request etc. The marking processing 1002 sets up condition directions of the beginning all the server hierarchy's control objects. The read-out processing 1004 reads the received condition structure which can be conveyed, and extracts status information and a corresponding property type.

[0086] The check processing 1006 performs the integrity check of the condition structure which can be conveyed (using for example, a ##VIEWSTATEMAC value). With an operation gestalt with this invention, it judges whether the;(3) receiving integrity code which is contained in the condition structure which reads a receiving integrity code from the condition structure which can be (1) conveyed, and which can be;(2) conveyed and which calculates the integrity code of itself from status information at least compares with the integrity code which newly calculated, and the condition structure which can be conveyed is altered between round trips of a client in check processing 1006. When the condition structure which can be conveyed is judged to be altered, a server side code can answer by continuing or considering processing without the status information which closes load processing as an error etc. so that this exception may be treated.

[0087] The inspection processing 1008 inspects the file check code contained in the condition structure which can be conveyed. With the operation gestalt of this invention, in the inspection processing 1008 (1) The file check code which carried out;(2); (3) reception which reads the file check code which received from the ##VIEWSTATE field, and which calculates the file check code of itself from the contents of the ASP+ resource memorized by the server at least Comparing the newly calculated file check code, the status information of the condition structure which can be conveyed judges whether it corresponds to the same version of the ASP+ resource of a server. It checks not changing between the round trips of the ASP+ resource on a server of the condition structure to a client which can be conveyed using the inspection processing 1008. When the ASP+ resource on a server changes, a server cancels the received status information, and closes load processing, and sends an error to/or a client. With a certain operation gestalt, the control object hierarchy of the first condition is created, and before a server continues the processing to a request, load processing is closed. The traverse processing 1010 traverses a server side control object hierarchy's page control object.

[0088] The syntax-analysis processing 1012 analyzes the syntax of the status information of the control

object hierarchy corresponding to the control object of the insert-die ##VIEWSTATE field. The control object which corresponds is a page control object traversed in the traverse processing 1010 1st repeatedly. Repeatedly [of a degree], the control object which corresponds is a descendant control object of a page control object, and the traverse processing 1028 clarifies the whereabouts. The syntax-analysis processing 1012 extracts status value, and when analyzed at the beginning, it can change into a given property type the value which was a string format. The load processing 1014 loads the status information the syntax of was analyzed from the ##VIEWSTATE value to the property of a control object. The directions processing 1016 sets up change condition directions of a current control object. By saving behind only the condition of a control object that the condition changed from the original condition, these directions can be used in order to make into min size of the condition structure which can be conveyed. Or all status information may be saved. It means that change which it changed into the condition of a control object had also set up change condition directions of each control object during handling of postback data, handling of a postback event, and data-coupling processing on a control object hierarchy. However, with an operation gestalt with this invention, any data change directions set up for data-coupling processing are reset so that the condition of the property may not be recorded on the condition structure which can be conveyed, and the size of the condition structure which can be conveyed is further decreased by it. It depends for the condition of such a property on the data specified with the joint relation updated by the data-coupling processing 512 of drawing 5 .

[0089] In the decision processing 1018, when additional status information exists in a current control object, processing progresses to 1022. When it exists, in processing 1022, the syntax of the status information of the following control object property in a current control object is analyzed, and processing progresses to 1014. In the decision processing 1018, when there is no additional status information in a current control object, it progresses to the decision processing 1020, and when other control object status information cannot be found in a ##VIEWSTATE value here, it progresses to processing 1024. In the decision processing 1020, when there is a control object of the addition by which the condition is not processed, it progresses to processing 1026 and the syntax of the status information of the following control object is analyzed from a ##VIEWSTATE value here. In the traverse processing 1028, the following control object corresponding to the new status information the syntax of was analyzed by processing 1026 is traversed. Then, it progresses to the load processing 1014. With an operation gestalt with this invention, repetitive syntax-analysis processing of drawing 10 performs processing in which it does not serialize.

[0090] Drawing 11 shows the process flowchart which saves the status information from a control object hierarchy's control object to the condition structure which is transmitted to the client in an operation gestalt with this invention, and which can be conveyed. The initialization processing 1100 initializes the condition structure which can be conveyed. Computation 1102 calculates a file check code based on the current version of an ASP+ resource. With the operation gestalt of this invention, a file check code is based on the hash algorithms (for example, MD5 etc.) of the contents of an ASP+ resource of only hashing the component with which some of contents of the file or files were specified. The load processing 1104 loads the file check code calculated by the condition structure which can be conveyed. The traverse processing 1106 traverses a control object hierarchy's page object.

[0091] In the operation gestalt of this invention, a server side control object can be created using the "MaintainState" property which shows whether the property value of a control object (and the child object) should be saved in the condition structure which can be conveyed. When it is not necessary to save, the traverse processing 1106 can skip a server side control object and its child object in this process. That is not right, and if the condition of a control object should be held according to a "MaintainState" property, the traverse processing 1106 traverses a control object.

[0092] Processing progresses to the read-out processing 1115 by the decision processing 1108. Here, new status value and its type are read from a control object property. Processing 1110 records new status value and its type, and progresses to processing 1111. A property is expressed with the "null" value of the ##VIEWSTATE field when a property does not exist in a current control object for this explanation. If the property of a current control object is still the original condition in processing 1108, it will

progress to the decision processing 1111. When another property exists in a current control object in processing 1111, it progresses to the decision processing 1108.
[0093] When other properties do not exist in a current control object in processing 1111, it progresses to the decision processing 1112, and when another control object exists in a hierarchy, it progresses to the traverse processing 1114 here. The traverse processing 1114 traverses the following control object in a control object hierarchy, progresses to the decision processing 1108, and accesses the property of a control object. In processing 1112, when other control objects do not exist in a hierarchy, the new status information (a status value type is included) acquired by previous processing is serialized, and it progresses to the serialization processing 1116 which memorizes them in the condition structure which can be conveyed. Processing 1118 calculates an integrity (using MD5 algorithm) code, and processing 1120 memorizes an integrity code in the condition structure which can be conveyed for example, in the ##VIEWSTATEMAC field. Integrity code processing and field check processing are arbitrary, and can be excluded with another operation gestalt of this invention. Processing 1122 transmits the condition structure which can be conveyed to a client at the HTTP request containing the HTML code which was generated by for example, the control object hierarchy and by which the rendering was carried out etc.
[0094] The operation gestalt of this invention in this specification is performed as a logic step of one or more computer systems. Logic processing of this invention is performed as a machine module with which it interconnected within the computer system beyond (2) 1 ** as a processor execute step sequence by which executive operation is carried out by the computer system beyond (1) 1 **. Activation is the problem of selection and it depends for it on the performance requirements of the computer system which performs this invention. Therefore, the logic processing which constitutes the operation gestalt of this invention of a publication on these specifications can be variously expressed as processing, a step, an object, or a module.
[0095] A specification, an above-mentioned example, and above-mentioned data offer the structure of the operation gestalt of this invention, and perfect explanation of use. Since this invention can be carried out with many gestalten, without deviating from the pneuma and the range of this invention, this invention is in an attached claim.
[0096]
[Effect of the Invention]

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] Generally especially this invention relates to management of the condition of the server side control object which processes the client side user interface element of a web page about a web server framework.
[0002]

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] A typical web browser receives data from the web server which specifies the appearance of a web page and basic actuation which are displayed in a client system. As a typical procedure, a user specifies the uniform resource (resource) locator (henceforth "URL") which is the global address of the resource on World Wide Web, and a desired website is accessed. Generally, the vocabulary "a resource" is the data which can be accessed by the program, or a routine. An example of URL is "HYPERLINK "http://www.microsoft.com/ms.htm" http://www.microsoft.com/ms.htm." The 1st part of this example of URL shows the given protocol (for example, "http") used for a communication link. The 2nd part specifies the domain name (for example, "HYPERLINK "http://www.microsoft.com" www.microsoft.com") which shows the whereabouts of a resource. The 3rd part specifies the resource in a domain (for example, file called "ms.htm"). This is followed and they are a browser and HYPERLINK "http://www.microsoft.com" www.microsoft.com. The HTTP (hypertext transport protocol) request relevant to the example of URL for taking out the data relevant to the ms.htm file in a domain is generated. The web server which is acting as the host of the www.microsoft.com site returns reception, the demanded web page, or a resource to a client system by the HTTP response, and displays a HTTP request on a browser.
[0003] The "ms.htm" file of the above-mentioned example contains the static HTML (HyperText Markup Language) code. HTML is a plane (plaintext) text authoring language used for the document (for example, web page) creation on World Wide Web. Since it is such, an HTML file is taken out from a web server and can be displayed on a browser as a web page. A developer can specify the text and list which are displayed on a browser and by which formatting was carried out, form, a table, a hypertext link, an in-line image, voice, and background graphics using HTML, and a graphical experience which it comes to expect while the user is looking at the information from the Internet can be offered. However, an HTML file is a static file which is not supporting dynamic generation of web page contents in essence.
[0004] When displaying dynamic contents, such as change of a stock price, and traffic information, the server side application program for handling the dialogue between usually more complicated client-server is developed. A server side application program processes a HTTP request, and generates the suitable HTML code for transmission of the HTTP response to a client. An example of a HTTP request may contain parameters, such as data from the data or the web based form in an enquiry string. Since it is such, a server side application program can process this parameter, and can generate the HTML code dynamically by the HTTP response to a client. The document containing the suitable HTML code may be generated dynamically, using the sequence of one or more text write-in processings to a memory device by which formatting was carried out as an example of a server side application program. Then, it is transmitted to a client system by the HTTP response, and the obtained document is displayed on a browser as a web page there.
[0005] Development of a server side application program is a complicated activity as which it is required it is not only well versed in the usual HTML coding used for a web page design, but that it is well versed in pro GURAMUBE six including one or more programming language (for example, C++,

Perl, Visual Basic, or Jscript). However, a web page designer is graphic designer or an editor in many cases, and may be inexperienced in a program. Furthermore, if complicated web page development is simplified, the rate of development of new web contents can be gathered by any developers. Generally, a great effort is required and development of a custom-made server side application program is also actually like [which a developer regards as not wanting to often try it]. Therefore, it is desirable to offer the development framework to which a developer can create and process a web page dynamically by the minimum program.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention]

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] One means which makes min the requirements for a program of dynamic web page generation is an active server page (ASP) framework offered by Microsoft Corp. An ASP resource processes typically the HTTP request which specifies an ASP resource as a desired resource, after that, generates the HTML code as a result of the HTTP response to a client, for example, contains Visual Basic or Jscript. Furthermore, a page developer does not require that all components should be written in from a scratch (from a condition without anything), and or the ASP resource was developed beforehand, refer to a third person's client side library component (for example, client side "ACTIVEX" control) for it. However, in the present server side application framework, programming which must manage dynamically client side user interface elements (for example, a text box, a list box, a carbon button, a hyperlink, an image, voice, etc.) within server side application needs a still advanced programming technique and a considerable effort. It is that an unsolved technical problem processes a user interface element so that a web page developer can focus on other aspects of a web page about property capsulation programming demanded.
[0007] However, server side processing of a client side user interface element may be typically accompanied by the complicated status management problem in a server side application program. The condition of the server side processing module corresponding to a client side user interface element expresses the property and configuration of a server side module. However, in the communication link between a client and a server, since the communication link connection between a client and a server is not expected but it may be cut, with a client / server model without status management, it will point so that a server may not maintain the condition of the process generated between a client and a server.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] According to this invention, **** and other technical problems are solved by offering the status management by which server side processing was encapsulated about the client side user interface element with which the server does not hold the condition between two requests. This is related with the condition (for example, "view state" (viewstate)) of one or more server side control objects of a control object hierarchy by processing of a client side user interface element including status management processing. A server side control object processes and generates the client side user interface element displayed on a web page. The hierarchy of a server side control object collaborates and generates an authoring language code a result like [again] the criterion HTML displayed on the web page of a client. As long as a client supports for example, the criterion HTML or another authoring language, it may be what kind of browser.
[0009] The status information of a server side control object has between a client and a server conveyed between two HTTP requests with the condition structure which can be conveyed in the client / server model which does not perform status management rather than is held at a server side (in order to fill). If a server receives a HTTP request from a client, a server will extract status information from the condition structure from a HTTP request which can be conveyed, and will distribute this status information to each suitable control object of a server side hierarchy. The integrity of the condition structure which can be conveyed is also inspected in relation to the condition structure which can be conveyed using the coded structure which was generated from status information.
[0010] The status management approach of the server side control object corresponding to the client side user interface element built into the web page displayed on a client is offered. A server side control object is created to a control object hierarchy, and a client side user interface element is processed. The condition structure which can be conveyed is received from a client. The condition structure which can be conveyed includes the status information which shows the status value of a control object hierarchy's server side [at least one] control object. Status information is extracted from the condition structure which can be conveyed. When the status value from status information relates to the server side control object, the status value is loaded to the property of a server side control object.
[0011] The computer program product which manages the condition of two or more server side control objects corresponding to two or more client side user interface elements built into the web page which is created by the control object hierarchy and displayed on a client is offered. The condition structure which was received from the client and which can be conveyed includes the status information relevant to one or more server side control objects of a control object hierarchy. Status information is serialized and the hierarchy of the property of status value, the related property data type, and a server side control object is extracted. Based on a hierarchy, the server side control object in a control object hierarchy is discovered. Status value is loaded to the property of a server side control object.
[0012] The product as a computer program product is offered. According to a computer system, the operation gestalt with the computer program product by this invention can be read, and contains the computer program storage which codes the computer program which performs the computer process which manages the condition of the server side control object corresponding to the client side user

interface element built into the web page displayed on a client. Shape is taken by the computer system by the subcarrier and another operation gestalt of the computer program product by this invention includes the computer data signal which codes the computer program which manages the condition of the server side control object corresponding to the client side user interface element built into the web page displayed on a client. The product generated by the process of this invention is offered as an authoring language code which is sent to a client and includes the condition structure which can be interpreted by the browser on a client, and which can be conveyed.

[0013]

[Embodiment of the Invention] In an operation gestalt with this invention, on a web server, web page contents are generated dynamically and displayed by the client. A client and a web server communicate by the network using for example, a HTTP request and a HTTP response. Thus, a web server generates web page contents, such as the HTML code gestalt, and transmits the contents to the client which can display a web page on a browser. The server side control object corresponding to each user interface element and logic target of a web page is created on a web server, and processes and generates the web page contents used by the client side browser which displays and processes a web page. A server side control object is declared in dynamic contents files, such as an ASP+ resource processed by the page factory which creates the hierarchy of a server side control object. The control object in a hierarchy collaborates and processes the request received from the client, subsequently, it generates result web page contents, transmits to a client, and terminates the control object in a hierarchy.

[0014] In an operation gestalt with this invention, a page object can be illustrated as a control object hierarchy's best level. In the page object which is also a control object, each child control object spreads to the hierarchy of a large number level including one or more child control objects of itself including one or more child control objects typically. A page object and a descendant's control object perform a processing sequence, and process and generate the web contents corresponding to a client side user interface element.

[0015] One of the processings in this sequence includes the load processing which un-serializes the status information received with the condition structure which is included in a request from a client, and which can be conveyed to suitable one or two or more suitable objects in a hierarchy. The traverse processing in this load processing traverses a control object hierarchy according to the hierarchy of the condition structure which can be conveyed, in order to find a control object suitable in a hierarchy. in an operation gestalt with this invention, the received status information includes only the condition of the control object which changed from the initial state (namely, this status information -- difference -- only data are included). With another operation gestalt, if it will not be to change, there is a thing including some another combination of all the conditions of all the control objects in a hierarchy or one or more status information of a server side control object, and a hierarchy.

[0016] Another processing of the above-mentioned sequence includes the preservation processing which serializes the status information from the control object from which the condition changed. Status information adds status information to the condition structure which is transmitted to a client and which can be conveyed. A client returns the structure which can be conveyed to a web server at the next request to a web server. If the condition structure which can be conveyed is received, a control object hierarchy will be re-created and a hierarchy condition will return to the hierarchy condition of a former response by load processing.

[0017] It can avoid transmitting the condition structure which can be conveyed to a client in another operation gestalt. The condition structure which can be conveyed remains in a server, without being transmitted, and may be transmitted to resource somewhere else (for example, another server). For example, in the case of load balancing, the 1st server communicates with a browser by the 1st HTTP transaction, but the 2nd server can communicate with a browser by the 2nd HTTP transaction. Therefore, the condition structure which can be conveyed is sent to the 2nd server in which the 2nd carries out HTTP transaction processing from the 1st server, and can bypass a client by it.

[0018] Drawing 1 shows the web server which generates dynamically the web page contents displayed on the client in an operation gestalt with this invention. A client 100 performs the browser 102 which

displays the web page 104 on the indicating equipment of a client 100. A client 100 may also include the client computer system which has a display like a video monitor. The "INTERNET EXPLORER" browser currently sold by Microsoft Corp. is an example of the browser 102 in an operation gestalt with this invention. As an example of other browsers, it is "NETSCAPE NAVIGATOR". It reaches. Although there is "MOSAIC" etc., it does not restrict to this. The text box control 106 and two carbon button control 108 and 110 are included in the illustrated web page 104. A browser 102 can receive the HTML code from a web server 116 by the HTTP response 112, and displays the web page described by the HTML code. Although HTML is explained with reference to a certain operation gestalt Not the thing restricted especially but SGML (Standard Generalized Markup Language), XML (eXtensible Markup Language) -- and It is the markup language of the XML base. It is thought that other authoring languages containing WML (Wireless Markup Language) designed so that the content and user interfaces of narrow-band radio equipment, such as a pocket bell (trademark) and a cellular phone, might be specified are within the limits of this invention. Furthermore, although the criterion HTML 3.2 is mainly indicated in this description, any versions of HTML may be contained within the limits of this invention.

[0019] The communication link with a client 100 and a web server 116 can be performed using a series of processings of the HTTP request 114 and the HTTP response 112. Although HTTP is explained with reference to a certain operation gestalt, it is not restricted especially and it is thought that other transport protocols including S-HTTP are within the limits of this invention. In a web server 116, the HTTP pipeline module 118 receives the HTTP request 114, analyzes URL, and calls the suitable handler which processes a request. The web server 116 is equipped with two or more handlers 120 treating the resource of a different type in the operation gestalt with this invention.

[0020] For example, when URL specifies a static contents file 122 like an HTML file, a handler 120 accesses the static contents file 122, and sends the static contents file 122 to a client 100 by the HTTP response 112 through the HTTP pipeline 118. Or in an operation gestalt with this invention, when URL specifies a dynamic contents file 124 like an ASP+ resource, a handler 120 accesses the dynamic contents file 124, processes the contents of the dynamic contents file 124, and generates the HTML code the result for web page 104. In an operation gestalt with this invention, the result HTML code contains standard HTML3.2 code. Generally, a dynamic contents file is the server side declaration data storage section (for example, ASP+ resource) which can use the authoring language which describes the web page which should be displayed on a client for generating dynamically. And the HTML code for web pages passes the HTTP pipeline 118, and is sent to a client 100 by the HTTP response 112.

[0021] During this processing, the handler 120 was beforehand developed again, in order to simplify a development effort, or it can access the library of the 3rd person code. One of such the libraries is the server side class control library 126, and a handler 120 can illustrate from here the server side control object which generates HTML data as a result of processing a user interface element and displaying on a web page. in an operation gestalt with this invention, one or more server side control objects are visible on the web page described by the dynamic contents file 124 -- it hides-like and maps to one or more user interface elements.

[0022] On the other hand, the 2nd library is a client side control class library 128 like the library containing the "ACTIVEX" component from Microsoft Corp. "ACTIVEX" control is a COM (component object model) object which follows a fixed criterion in the method of a dialogue with a client and other components. Client side "ACTIVEX" control is the component of the COM base as for which downloads to a client automatically and executive operation may be carried out to it by the web browser of a client. A server side ACTIVEX component (not shown) is a component of the COM base which may be performed on a server, in order to achieve the various server side functions in which the server side functionality of stock price retrieval application or a database component is offered. ACTIVEX is indicated by the detail by "understanding of ACTIVEX and OLE" (David Chappelle, the Microsoft press, 1996).

[0023] In contrast with "ACTIVEX" control, the server side control object of the operation gestalt of this invention specified as the dynamic contents resource 124 corresponds to the user interface element

built into the web page on a client logically. A server side control object can generate the effective HTML code which may contain the locator which carries out the reference of the given client side "ACTIVEX" control to for example, a HTML tag again. When the browser has already had the code for client side "ACTIVEX" control in the storing system, executive operation of the "ACTIVEX" control is carried out within the web page on a client. If it becomes right [that], a browser will download the code for "ACTIVEX" control from the resource specified by the locator, and will carry out executive operation of the "ACTIVEX" control within the web page on a client. The server side control object in the operation gestalt of this invention can raise an event again to the server side "ACTIVEX" control used for performing stock price retrieval application on a server.

[0024] A handler 120 accesses one or more non-user interface server components 130 which carry out executive operation on a web server 116 or another accessible web server again. A non-user interface server component 130 like stock price retrieval application or a database component is referred to in the dynamic contents file 124 processed by the handler 120, or is related with it. The server side event raised by the control object declared by the dynamic contents file 124 may be processed in server side code which calls the suitable method of the non-user interface server component 130. Consequently, the processing offered by the server side control object can simplify programming of non-user interface server component stereo-NENTO 130 by encapsulating processing and generation of a web page of a user interface element, and, thereby, the developer of the non-user interface server component 130 can be concentrated on development of the function of the application proper instead of a user interface problem.

[0025] Drawing 2 shows the flow chart of processing of the client side user interface element using the server side control object in an operation gestalt with this invention, and generation processing. In processing 200, a client transmits a HTTP request to a server. A HTTP request contains URL which specifies resources, such as an ASP+ resource. In processing 202, a server calls the suitable handler which receives a HTTP request and processes the specified resource. A HTTP request includes the condition structure including a hierarchy which can be conveyed in one or more the status information and arbitration about a server side control object. However, the HTTP request of the beginning to a given page does not include typically the condition structure which can be conveyed from the condition not changing in the server of the given page. Furthermore, in case the condition structure which can be conveyed checks the effectiveness of property data-type information and status information (that is, it is checked that status information has not been altered by the client), it may also contain the integrity code for assisting a server. Processing 204 generates a server side control object hierarchy based on the content of the specified dynamic contents file (for example, ASP+ resource). Reading appearance of the ASP+ resource is carried out in processing 203. In processing 205, the status information received with the condition structure which can be conveyed is loaded to the suitable server side control object in a hierarchy, and a control object is returned to a former condition.

[0026] In processing 206, a control object hierarchy's server side control object performs postback event handling, postback data handling, status management, and one or more processings in data coupling. From a client, the postback event and data (collecting "postback input") from a user interface element are sent to a server, and are processed. For example, especially a postback event is not restricted and may also contain the "data change" event from the client side text box element sent to "a mouse click" or a server from a client side carbon button element. Especially postback data are not restricted and may also contain the text inputted into the index of the item chosen from the text box element or the drop down box by the user.

[0027] In processing 209, a property value, related type information, and a hierarchy are stored in the condition structure which is transmitted to a client and which can be conveyed. In processing 208, in order that each server side control object in a hierarchy may generate authoring language data like the HTML code for the display by the web page of a client side user interface element (or rendering), it is called. Although the vocabulary "a rendering" may mean the processing which displays graphics on a user interface, in this description, the "rendering" also means generation processing of the authoring language data which may be interpreted by client application like the browser for a display and a client

side function. The rendering processing 208 generates the authoring language data showing the condition structure which can be conveyed again. More detailed explanation of processing 206 and the rendering processing 208 is given in connection with drawing 6 . In a certain operation gestalt, the call of the render() method in each control object is performed using a tree traversal sequence. That is, the call of the render() method of a page object becomes the repetitive traverse covering the suitable server side control object in a hierarchy. As an option which calls the render() method of a suitable control object, approaches, such as event signaling or the object registration technique, may be used. A parenthesis specifies the "render()" level which shows the method of comparing with a data value.
[0028] In an operation gestalt with this invention, actual creation of each server side control object may delay until a server side control object is accessed in processings 206 or 208 (handling of a postback input, loading of a condition, rendering of a control object to the HTML code, etc.). When a server side control object is not accessed for a given request, server processing is optimized by delaying creation of a control object and eliminating unnecessary control object creation processing.
[0029] In processing 210, authoring language data (for example, the HTML code) including the condition structure which can be conveyed are transmitted to a client by the HTTP response. In processing 214, a client receives the HTML code including the condition structure relevant to the web page which should be displayed which can be conveyed. The condition structure which can be conveyed may be saved at a client, in order to return to a server at the next HTTP request. In processing 216, a client system incorporates the user interface element of a new page according to the HTML code received from the HTTP response (for example, it displays). however, a user interface element -- incorporating -- he should also understand that you may include non-display processing of control of read-out to the memory which offers voice or a tactile-sense output and writing, and script processing etc. A server side control object hierarchy is ended in processing 212. In an operation gestalt with this invention, the server side control object in a hierarchy answers the HTTP request which refers to the associated ASP+ resource, is created, and after the rendering of authoring language data (for example, HTML data) finishes, it is destroyed. In another operation gestalt, processing 212 may be performed before processing 210 after processing 208.
[0030] Drawing 3 shows an example of the module in the web server used in an operation gestalt with this invention. A web server 300 accepts the HTTP request 302 in the HTTP pipeline 304. The HTTP pipeline 304 may also contain various modules, such as logging of web page statistics, user collating, a right of user access, and a module for the formation of an output cache of a web page. Eventually, each input HTTP request 302 received by the web server 300 is processed by the specific instance of an IHTTP handler class (it illustrates as a handler 306). A prefix called IHTTP shows that the class is the interface of a HTTP handler. A handler 306 analyzes a URL request and calls a suitable handler factory (for example, page factory module 308).
[0031] In drawing 3 , the page factory 308 related with the ASP+ resource 310 is called, and the instantiation and configuration of an object which were declared in the ASP+ resource 310 are handled. In a certain operation gestalt, an ASP+ resource may be recognized or referred to by specifying a specific suffix (or a file extension part like ".aspx") as a file. If the request to given ".aspx" resource is first received by the page factory module 308, the page factory module 308 will search a file system, and will obtain a suitable file (for example, .aspx file 310). This file may also contain the text (for example, authoring language data) which may be behind interpreted or accessed by the server which processes a request, or the data (for example, cutting tool code data or coded data) in another format. When a physical file exists, the page factory module 308 reads a file to an aperture, and reads the file to memory. When a file is not found, the page factory module 308 returns the suitable error message "a file is not found."
[0032] After reading the ASP+ resource 310 to memory, the page factory module 308 processes a file content, and builds the data models (for example, a list [of script blocks], directive, static text area, and hierarchy server side control object, a server side control property, etc.) of a page. A data model is used for generating the source list of a new object class like the COM+ (Component Object Model+) class to which the class of the page base is made to extend. The page base class contains the code which

specifies the structure, property, and function of a basic page object. Subsequently, a source list is dynamically compiled by intermediate language. Intermediate language is COM+ IL. A code and Java A cutting tool code and Modula 3 A code and SmallTalk A general purpose or custom-made orientation linguistic code of a code, the Visual Basic code, etc. may also be included. In another operation gestalt, intermediate-language processing may be excluded and the instruction of a proper may be directly generated from a source list or a source file (for example, ASP+ resource 310). The control class library 312 may be accessed with the page factory module 308, in order to obtain the server side control class which is used for a control object hierarchy's generation and which was defined beforehand.

[0033] The page factory module 308 outputs the page object 314 which is a server side control object equivalent to the web page 104 of drawing 1 . The page object 314 and its child object (for example, the text box object 318, the carbon button object 320, and another carbon button object 322) are the control object hierarchy's 316 examples. The example of other control objects can be considered according to this invention, is not restricted especially, and contains the object and custom-made control object corresponding to HTML control of a table 1. The page object 314 corresponds to the web page 104 of drawing 1 logically. The text box object 318 corresponds to the text box 106 of drawing 1 . Similarly, the carbon button object 320 corresponds to the additional carbon button 108 of drawing 1 , and the carbon button object 322 corresponds to the deletion carbon button 110 of drawing 1 . The page object 314 relates to other control objects and hierarchy targets on a server. In a certain operation gestalt, a page object is a container object which contains the child control object hierarchical. The hierarchical relationship of other gestalten, such as a dependency, can be used with another operation gestalt. In a complicated control object hierarchy, one child object may be a container object of other child objects rather than it has the child object of a large number level.

[0034] It sets in the above-mentioned operation gestalt, the control object hierarchy's 316 control object is created and performed on a server 300, and each server side control object "carries out the mirror of the user interface element with which it corresponds on a client." The server side control object of this operation gestalt collaborates, handles the input from the HTTP request 302 again, manages the condition of a server side control object, performs data coupling with a server side database, and generates the authoring language data (for example, the HTML code) used for the display of a web page as a result of a client. Result authoring language data are generated from the server side control object hierarchy 316 (namely, rendering), and are transmitted to a client by the HTTP response 324. For example, the result HTML code can embody any effective HTML configurations, and in addition to this, control of an ACTIVEX type, a JAVA (trademark) applet, a script, and when it is processed by the browser, refer to any web resources which produce client side user interface elements (for example, a control carbon button, a text box, etc.) for it.

[0035] By declaration made with the ASP+ resource 310, a server side control object can access one or more non-user interface server components 330 for a dialogue with the non-user interface server component 330 and a client side user interface element. For example, a postback input can be answered and a server side control object can raise a server side event to the non-user interface server component registered into those events. Thus, the non-user interface server component 330 minds a user interface element for a dialogue with a user, and it can perform it, without programming a code required displaying and processing these elements.

[0036] In short, the operation gestalt of this invention contains on a server creation and the server side control object by which executive operation is carried out, in order to generate the HTML code sent to a client. The HTML code can embody any effective HTML configurations, and is control of an ACTIVEX type and JAVA, for example. Any web resources of the others which generate the user interface carbon button and other user interface elements in an applet, a script, and a client can be referred to. The user in a client can have a dialog with these user interface elements that correspond to a server side control object logically, and can return a request to a server. A server side control object is re-created on a server, and it processes the data, the event, and other properties of a user interface element so that the HTML code of the next round which should transmit to a client as a response may be generated.

[0037] If drawing 4 is referred to, an example of the computer system of the operation gestalt of this invention contains the general purpose computer equipment of the gestalt of the conventional computer system 400 containing the system bus 406 which connects the various system components containing the processor unit 402, a system memory 404, and a system memory 404 to the processor unit 400. System buses 406 may be any of the bus structure of some types containing the peripheral bus and local bus which use a memory bus or a memory controller, and various bus architectures. The system memory contains the memory (ROM) 408 only for playbacks, and random access memory (RAM) 410. The unformatted input / output system 412 (BIOS) containing the basic routine which helps a transfer of the information between the elements within a computer system 400 are stored in ROM408.
[0038] The computer system 400 contains the optical disk drive 418 which performs read-out and the writing of a removable optical disk 419 like the magnetic disk drive 414 and CD ROM which perform further read-out and the writing of a hard disk drive 412 and the removable magnetic disk 416 which perform read-out and the writing of a hard disk, DVD, or other optical media. The hard disk drive 412, the magnetic disk drive 414, and the optical disk drive 418 are connected by the hard disk drive interface 420, the magnetic disk drive interface 422, and the optical disk drive interface 424 system bus 406, respectively. A drive and its medium which can be related computer read offer the instruction of a computer system 400 which can be computer read, DS, a program, and the non-volatile storage section of other data.
[0039] Although the hard disk, the removable magnetic disk 416, and the removable optical disk 419 are used for this description in the above-mentioned environmental example of a publication, the medium type [other] in which data storage is possible and which can be computer read can be used for the above-mentioned example of a system. The medium of other types of these which can be used for the above-mentioned example of operating environment which can be computer read has for example, a magnetic cassette, flash memory card, a digital videodisc, a BERUNUI (Bernoulli) cartridge, random access memory (RAM), the memory (ROM) only for playbacks, etc.
[0040] Many program modules may be stored in a hard disk, a magnetic disk 416, an optical disk 419, and ROM408 or RAM410, and these contain the application program 428, other program modules 430, and the program data 432 of 426 or 1 or more operating systems. A user can input a command and information into a computer system 400 with input units, such as a keyboard 434 and a mouse 436, or other pointing equipments. As other input devices, there are a microphone, a joystick, a gamepad, a satellite dish, a scanner, etc., for example. These and other input devices are connected to the processor 402 through the serial port interface 440 connected to the system bus 406 in many cases. However, these input devices may be connected by other interfaces, such as a parallel port, a game port, or Universal Serial Bus (USB), again. The monitor 442 or the indicating equipment of other types is also connected with the system bus 406 through the interface of a video adapter 444 etc. In addition to a monitor 442, typically, a computer system contains other circumference output units (not shown), such as a loudspeaker and a printer.
[0041] A computer system 400 can operate in the environment using the logical connection to one or more remote computers like a remote computer 446 connected by network. a remote computer 446 -- a computer system, a server, a router, Network PC, and a pier (peer) -- it may be equipment or other common network nodes, and there are many elements typically mentioned above in connection with a computer system 400, or all are included. Network connection contains Local Area Network (LAN) 448 and Wide Area Network (WAN) 450. Such a network environment is not new in office, an enterprise magnitude computer network, intranet, and the Internet.
[0042] When using by the LAN network environment, a computer system 400 is connected to a local network 448 through a network interface or an adapter 452. When using by the WAN network environment, a computer system 400 includes the modem 454 or other means for establishing the communication link by Wide Area Network 450 like the Internet typically. Built-in or external any is sufficient as a modem 454, and it is connected with the system bus 406 through the serial port interface 440. In the environment connected by network, the program module described in relation to the computer system 400 or its part may be memorized by remote memory storage. The illustrated network

connection is an example and can use other means for the communication link establishment between computers.
[0043] In the operation gestalt of this invention, a computer 400 expresses a web server and CPU402 carries out executive operation of the page factory module on the ASP+ resource memorized by at least one of storages 416, 412, 414, 418, and 419 or memory 404. A HTTP response and a request communicate by LAN448 connected to the client computer 446.
[0044] Drawing 5 is a process flowchart showing server side processing of the page object in an operation gestalt with this invention, and other control objects. In processing 500, the page object construction section is called with the page factory module 308 (refer to drawing 3). Consequently, it is created in order that a page object (see the page object 314 in drawing 3) "may carry out the mirror of the web page user interface element on a client." In processing 502, a page factory module calls the ProcessRequest member function of the page object which begins to process [gradual] the HTTP request received from the client. In the 1st phase of 1 operation gestalt of this invention, server side creation processing (not shown) may be creation of the descendant server side control object contained in the control object hierarchy of a page object. That is, in order that the construction section of a child control object may create the control object between the processing life times of HTTP request processing, call appearance is carried out repeatedly.
[0045] However, in another operation gestalt, creation of a child control object is delayable until a control object is needed for a given processing step (for example, rendering of the HTML code of the user interface element with which handling of a postback event, handling of postback data, loading of a view state and preservation, and data coupling analyze or correspond). Since the operation gestalt of the latter "which delays control object creation" can reduce utilization of unnecessary CPU and memory, it is the optimal. For example, there is a case where it will be called creation of the web page from which the user input event received from the client completely differs. In this case, a control object hierarchy makes it end promptly, and only in order to process the event which will illustrate a new different control object hierarchy of a new page, it is not necessary to illustrate the whole control object hierarchy of a former page.
[0046] The server call of the ProcessRequest method of a page object can be answered, and executive operation of the processings 504-520 can be carried out by the data page object and each descendant control object according to the data of a given HTTP request etc. In an operation gestalt with this invention, processings 504-520 are performed to each object of each in the sequence of drawing 5 . However, depending on a HTTP request, about given processing of another object, it may not be carried out in order or, as for the given processing to one object, processing may not be performed at all. For example, the 1st object performs the initialization processing 504 and its load processing 506, the postback data processing 508 is started and a descendant control object performs the initialization processing 504 and the load processing 506 of itself by overdue control object creation after that. Although the sequence of processing by the page object and the descendant control object is not restricted to this, it depends on the various factors [whether the current condition of the property of the data in a HTTP request, a control object hierarchy's configuration, and a control object and control object creation are performed behind time] to include.
[0047] The initialization processing 504 initializes a control object, after a control object is created by carrying out executive operation of the server side code of the arbitration about initialization in a dynamic contents file. Thus, each server side control object may be customized by the specific server side function declared by the dynamic contents file. In the operation gestalt of this invention, it was meant that a dynamic contents code customizes or extends the base page control class declared by the page developer with the ASP+ resource on a server. Compile of an ASP+ resource contains the declared code in the suitable initial code (for example, initialization () method of a page object and a descendant control object). The initialization processing 504 carries out executive operation of this code, and customizes or extends a page base class and the base class of a descendant control object.
[0048] In an operation gestalt with this invention, the status management of a server side control object is supported by the load processing 506 and the preservation processing 516 in which the condition

structure which can be conveyed is used, in order to contain the non-statement model of a client/server system by returning a server side control object to a former condition. With a certain operation gestalt, although a condition goes and comes back to a server in the one or more HTML fields in which the HTTP request / response of a couple hid, it is considered that other condition structures which can be conveyed, such as Cookie and the visible field, are within the limits of this invention.
[0049] In a series of processings by the given present request and present given response about a page between a client and a server, the condition of one or more control objects is recorded on the condition structure which can be conveyed by the preservation processing 516 after processing of a previous request. In an operation gestalt with this invention, the status information of the addition containing the control object identifier which enables a hierarchy or a server to associate a suitable control object and a given condition is also included in the condition structure which can be conveyed. In the next HTTP request, status information is returned to a server with the condition structure which can be conveyed. A server extracts status information from the received condition structure which can be conveyed, loads condition data to the suitable control object in a control object hierarchy, and returns each control object to the condition that it existed before the previous HTTP response. After the processing to the present request, again, the condition of one or more server side control objects is recorded on the condition structure which can be conveyed by the preservation processing 516, and returns the condition structure which can be conveyed to a client by the next HTTP response by it.
[0050] Each server side control object is changed into the same condition as the condition before a previous HTTP request as a result of the load processing 506. For example, when a text box control object includes a property value equivalent to "JDoe" before a previous HTTP response, the load processing 506 returns the same control object to a previous condition by loading text string "JDoe" to the property value etc. Furthermore, whether the condition of a given object was memorized and recovered can also constitute.
[0051] When 1 operation gestalt of this invention is summarized, "it is saved" after the condition of one or more server side control objects processing. The saved status information is transmitted to a client by the response. A client returns the status information saved by the next response to a server side. A server loads status information to the newly illustrated server side control object hierarchy so that a hierarchy's condition may return to a former condition.
[0052] With another operation gestalt, status information can be held for some other accessible web locations by the server during round trip on a server of returning from a server to a client and a server. After a server receives a client request, this status information is taken out by the server and may be loaded to the suitable server side control object in a control object hierarchy.
[0053] In processing 508, the postback data received from the HTTP request are processed. Postback data may be contained in the pay load of the HTTP request by the key value and hierarchical display (for example, XML) which became a pair, or other data display like RDF (Resource Description Framework). Processing 508 analyzes the syntax of a pay load, and identifies the identifier of the meaning of a server side control object. When an identifier (for example, "page1:text1") is found and the identified server side control object exists in a control object hierarchy, corresponding postback data are passed to the control object. For example, reference of drawing 1 sends the identifier of a meaning relevant to a text box 106 and text "JDoe" to a web server 116 with the pay load of the HTTP request 114. Processing 508 analyzes the syntax of the pay load of the HTTP request 114, and acquires the identifier and its related value (namely, "JDoe") of a meaning of a text box 106. And processing 508 analyzes the identifier of the meaning of a text box 106, identifies a corresponding server side control object, and passes a "JDoe" value to the object to process.
[0054] As the load processing 506 was explained, the property value of a server side control object may be returned to a former condition. If postback data are received, a server side control object will judge whether the property value of the point to which the passed postback value corresponds was changed. When it changes, logging of the change is carried out to the change list in which data change of a related control object is shown. After all postback data are processed within a control object hierarchy, a control object method is called and one or more postdata change events can be raised to one or more non-user

interface server components like stock price retrieval application started on a server. A postdata change event is an event which shows that for example, postback data changed the property of a server side control object. In an instantiation operation gestalt, such an event is sent to a system offer event queue, and can call the server component registered so that an event might be processed. Thus, a server side non-user interface server component can answer the event in which the trigger was carried out by change of the data of a server side control object. The option including using an application offer event queue, polling, and processing interruption which performs an event can also be considered within the limits of this invention.

[0055] A postback event is handled in processing 510. A postback event communicates with the pay load of a HTTP request. Processing 510 analyzes the syntax of the specific event target (for example, with the operation gestalt with this invention, label attachment is carried out with "##EVENTTARGET") which identifies the server side control object to which the event is turned. Furthermore, processing 510 analyzes the syntax of it, when there is a discovered event argument, and it gives it to the server side control object which had the event argument (for example, label attachment is carried out with "##EVENTARGUMENT" with the operation gestalt with this invention) specified. A control object starts the event processed with the non-user interface server component (for example, server side stock price retrieval application) relevant to a dynamic contents file.

[0056] Processing 512 solves data coupling between a server side control object and one or more databases with an accessible server. the property of a server side control object may be related with the property of a parent data-coupling container as shown in the table of a server side application database with an operation gestalt with this invention (or data coupling is carried out -- having). A page frame work piece can update the control object property which has the value of the parent data-coupling container property corresponding to between the data-coupling processings 612 and by which data coupling was carried out. Thus, the user interface element on the web page in the next response reflects the updated property value in accuracy. The control object property with which a user interface element corresponds is because it was automatically updated between the data-coupling processings 512. Similarly, a control object property may be updated by the parent data-coupling container field, and, thereby, updates a server side application database in the postback input from a server side control object again.

[0057] Processing 514 updates a large number by which executive operation may be carried out, before saving a control object condition and carrying out the rendering of the output. Processing 516 requires status information (namely, view state) from one or more control objects in a control object hierarchy, stores status information, and inserts it in the condition structure which is sent to a client with a HTTP response pay load and which can be conveyed. For example, a "grid" control object can save the present index page of the list of values, and a "grid" control object can return to this condition after the next HTTP request (namely, processing 506). As mentioned above, view state information expresses the condition of the control object hierarchy before the next action by the client (for example, before sending a HTTP response to a client). It will be used for changing a control object hierarchy into the condition of the point before client post back input process or data coupling if view state information returns.

[0058] The rendering processing 518 generates the suitable authoring language output (for example, HTML data) sent to a client by the HTTP response. A rendering is performed by top-down hierarchy tree WOKU and the embedded rendering code of all server side control objects. Processing 520 does the last clean-up activity (it is connection of a database in closing a file ****) of arbitration, and a control object hierarchy is terminated. Subsequently, processing performs return and processing 522 to 502, and a page object is ended by calling the destructive section there.

[0059] Drawing 6 shows an example of the part of the dynamic contents file (for example, ASP+ resource) in an operation gestalt with this invention. The line 1 of the ASP+ resource 600 is the initiation tag of an HTML file, and is declared as a literal in the ASP+ resource 600. A literal corresponds to the server side literal control object in a server side control object hierarchy. Since it is the 1st control object by declaration with the ASP+ resource 600, an index "0" is given to the literal control object

corresponding to a line 1. In rendering time amount, a literal control object only generates the new line for only containing in a "<html>" text and a HTTP response. The lines 2-12 of the ASP+ resource 600 express a code declaration block, and executive operation of this is carried out on a server (namely, "runat =server" of a line 2 it was shown by the attribute like). In the example of this ASP+ resource 600, instantiation of the server side control object in a control object hierarchy does not take place as a result of a code declaration block. That is not right, and by code declaration block, it "be connected with a wire" connect [with the server side control object declared with the line 15] Is sufficient for a server side code, or it is associated. The line 13 of the ASP+ resource 600 is the initiation tag of the text of an HTML file, is declared with the ASP+ resource 600 as a literal, and serves as a literal control object which has an index "1."

[0060] Declaration in a line 14 declares the server side form control object (index = "2") illustrated by the control object hierarchy. The tag in lines 17 and 18 is a termination tag expressed as the server side literal control object (index = "3") within a control object hierarchy. The termination tag of a line 16 ends the <form> declaration corresponding to a form control object.

[0061] On the following hierarchy level, declaration of a line 15 declares the server side label control object (index = "1") which has the identifier attribute ("id") of "Message" using HTML control tag "span". The index "0" of this hierarchy level and "2" are assigned to the literal control object (index = "2") of the white space literal a literal control object (index = "0") and next to the white space literals (for example, a tab, a new line, a tooth space, etc.) of precedence, respectively.

[0062] Drawing 7 shows a code, as a result of answering the partial example of the dynamic contents file of drawing 6 and being generated by one or more server side control objects. In order to simplify this explanation, only form, status management, and a span section are shown in the HTML code section 700. Although other tags like <html> and <body> are not contained in the HTML code section 700, the HTML code of these tags may be contained in 1 set of perfect HTML codes from the ASP+ resource of drawing 6 .

[0063] Lines 1 and 5 are initiation and the termination tags of the form section of the HTML code, and are generated by the server side form control object declared with the lines 14-16 of drawing 6 . The line 4 contains the HTML code of the label (namely, "Last Post Back:5/6/99") which shows the date of the last post back processing declared with the line 15 of drawing 6 .

[0064] The field in which lines 2 and 3 were hidden expresses 1 operation gestalt of the condition structure by this invention which can be conveyed. In a line 2, the status information of the server side control object declared with the ASP+ resource 600 is recorded on the field in which the name of "##VIEWSTATE" hid. A "##VIEWSTATE" field value is a text string showing the hierarchy of the control object in status value, a property type, and a control object hierarchy. Since load processing traverses a hierarchy and loads given status value to the property of the suitable control object in a hierarchy again, the hierarchy is included in the "##VIEWSTATE" field.

[0065] With an operation gestalt with this invention, a client has a dialog with a web server using a HTTP request / response pair sequence. Between the response to a client, and the next request which a server receives, a server does not need to maintain the condition of the server side control object relevant to connection with a given ASP+ resource or a given given client. Instead of maintaining, it is returned to a client with the condition structure which can be conveyed (by for example, HTTP response), and a given control object hierarchy's status information is returned by the server with delivery and the condition structure which can be conveyed (at for example, the next HTTP request).

[0066] A line 3 expresses the file in which the name of "##VIEWSTATE" which has the value "434333433" hid. "##VIEWSTATEMAC" is an integrity code in the condition structure which can be conveyed where it is used by load processing, in order to check that the ##VIEWSTATE value is not altered by the client. # At first, #VIEWSTATEMAC value is calculated by the server from the content of the ##VIEWSTATE value, and is sent to a client by the response. When a client returns the condition structure which can be conveyed, a web server calculates a new integrity code from the content of the received ##VIEWSTATE value. the ##VIEWSTATE value which the web server received when the received integrity code and the newly calculated code were equal -- validity -- or it is not altered -- **

(that is, the same as the ##VIEWSTATE value previously sent to the client) -- it judges. With an operation gestalt with this invention, an integrity code is calculated using an MD5 algorithm. MD5 is the algorithm created by Professor Donald RIBESUTO in 1991, in order to use it for creation of a digital signature. The hash function of an one direction, i.e., a message, is changed into reception, it changes it into a digital fixed string, and MD5 is also called a message digest. Such a method may also include the activity of the key of the secrecy which reduces the unjust actuation by an individual or a program without an access privilege. However, with other operation gestalten, you may also include another integrity coding technique containing the die length of generating the code based on MD4 algorithm or the hash algorithm of other arbitration, or the ##VIEWSTATE field.
[0067] The structure of the ##VIEWSTATE field of a line 2 expresses a control object and the hierarchical nest condition of those conditions (for example, status value and the property data type relevant to it) with an operation gestalt with this invention. # The tag used for an example of #VIEWSTATE field is shown in (a table 1). The tag containing "/" expresses a termination tag.
[0068]
[A table 1]

タグ	説明
<s>, </s>	ストリング値
<ax>, 	x 要素を有するアレイ
<i>	整数値
<hx>, </H>	x テーブルエントリを有するハッシュテーブル
<K>, </K>	ハッシュテーブルキー
<V>, </V>	ハッシュテーブル値
,	ブーリアン値
<d>, </d>	日付／時間値
<c>, </c>	カレンシ値
<A>, </A>	アレイリスト
<n>, </n>	ヌル値

[0069] Although an example of a coding implementation gestalt is shown in this description, with another operation gestalt, he should understand that other coding approaches can be used. For example, a hierarchy, status value, and those related property data types can also be specified using the hierarchy identifier of the meaning of each server side control object, or the data of the condition structure which can be conveyed can also be expressed using an XML related data format. Furthermore, using well-known encryption and well-known compression technology, safety can be secured and the size of the condition structure which can be conveyed can also be reduced.
[0070] Drawing 8 shows the control object hierarchy corresponding to the ASP+ resource of drawing 6 in an operation gestalt with this invention, and the ##VIEWSTATE field value of drawing 7 . Although it is not a control object hierarchy's control object component, it is shown as the best level of the hierarchy based on the ##VIEWSTATE field structure where the file check block 802 is shown in drawing 7 . More detailed explanation of the file check block 802 is given about drawing 9 . Or the file check block 802 is excluded or it is also thought that it is within the limits of this invention to make it together with the hierarchy level corresponding to the page control object 804.
[0071] The page control object 804 is illustrated as a control object hierarchy's best level, and corresponds to the special control object corresponding to the web page of a result itself. With an operation gestalt with this invention, especially the declaration (namely, declaration which has a "runat = server" attribute) for making the page object 804 illustrate is not needed.
[0072] In the following hierarchy level, the control objects 806, 808, 810, and 812 are contained in the

page object 804, or are illustrated as a child control object which has relation hierarchical. The index of a zero base is given to each control object of one hierarchy level based on the sequence to the bottom [from / when it can set to an ASP+ resource]. for example, -- a literal -- control -- an object -- a hierarchy -- 806 -- drawing 6 -- ASP -- + -- a resource -- 600 -- a line -- one -- it can set -- " -- < -- html -- > -- " -- a text -- and -- accompanying -- white space -- corresponding -- **** . Therefore, the index "0" is given to the literal control object 806. With the above-mentioned operation gestalt of this invention, the code declaration block extended from lines 2-12 does not become instantiation of a corresponding control object. a literal -- control -- an object -- 808 -- drawing 6 -- ASP -- + -- a resource -- 600 -- a line -- 13 -- it can set -- " -- < -- body -- > -- " -- a text -- and -- accompanying -- white space -- corresponding -- **** . Therefore, the index "1" is given to the literal control object 808. The form control object 810 supports the form declaration in the lines 14-16 of the ASP+ resource 600 of drawing 6 . Therefore, the index "2" is given to the form control object 810. The literal control object 812 supports the termination tag in the lines 17 and 18 of the ASP+ resource 600 of drawing 6 , and attendant white space. Therefore, the index "3" is given to the literal control object 812.

[0073] In the following hierarchy level, the control objects 814, 816, and 818 are contained in the form object 810, or are illustrated as a child control object which has relation hierarchical. The literal control object hierarchy 814 supports the white space before the span declaration in the line 15 of the ASP+ resource 600 of drawing 6 . Therefore, since the literal control object 814 is the 1st literal text (it is white space) which encounters in relation with this hierarchy level in an ASP+ resource, the index "0" is given to it. Similarly, the literal control object 818 supports the white space following the span declaration in the line 15 of the ASP+ resource 600 of drawing 6 . Therefore, the index "2" is given to the literal control object 818. The label control object 816 supports the declaration in the line 15 of the ASP+ resource 600 of drawing 6 . Therefore, the index "1" is given to the label control object 816.

[0074] Drawing 9 shows the nest version of the ##VIEWSTATE field value of drawing 7 . The nest version is indicated in order to simplify explanation of the hierarchical property of a ##VIEWSTATE field value into this description. The line 1 of the nest version 900 corresponds to the beginning of the file check block 802 which became the nest of drawing 8 . A line 1 specifies 2 element array and contains 2 element array of a line 3 as the string and the 2nd element of a line 2 as the 1st element. A line 2 specifies the string value showing a file check code. With an operation gestalt with this invention, a file check code is calculated as a hash value of the ASP+ resource (for example, ASP+ resource of drawing 6) of relation. The array which starts with a line 3 expresses the beginning 804 of the status information of the following hierarchy level, i.e., the page control object of drawing 8 . The zero value of a line 4 expresses the 1st element of 2 element array of a line 3 which can hold the status information about the property of page object itself. In the above-mentioned operation gestalt, since the status information of a page object is not recorded, the value is "null" (zero).

[0075] As shown in a line 5, the 2nd element of 2 element array of a line 3 shows the hash table including the status information of the child (namely, control objects 806, 808, 810, and 812 of drawing 8) of a page control object. The hash table of a line 5 contains one hash table entry indicated to be "1" to a tag <H1>. <K> of lines 6 and 8 and a </K> tag contain the integral value "2" shown in the line 7 equivalent to the key of a hash table entry, i.e., the index of the form control object 810 of drawing 8 . The ##VIEWSTATE value shown in drawing 9 does not include the status information of the literal control objects 806, 808, and 812 of drawing 8 . Or when other control objects or child control objects which saved status information exist in the hierarchy level same as a form control object 810, the hash table of this level may have one or more hash table entries for containing the control object of an addition of this level, and the key to each additional hash table entry may be the corresponding index of a control object.

[0076] A hash table entry value is included in <V> of lines 9 and 24, and a </V> tag, and contains another 2 element array which specifies the condition of a form control object and its child object. The 1st element of the array specifies the status information where the form control object (that is, expressed with the zero value in this operation gestalt) was saved.

[0077] The 2nd element specifies another 1 entry hash table showing the status information of the child

object of a form control object. <K> of lines 13 and 15 and a </K> tag contain the integral value "1" shown in the line 14 equivalent to the key of one hash table entry, i.e., the index of the label control object 816 of drawing 8 . The ##VIEWSTATE value shown in drawing 9 does not include the status information of the literal control objects 814 and 818 of drawing 8 .
[0078] A hash table entry value is included in <V> of lines 16 and 21, and a </V> tag, and contains another 2 element array which specifies the condition of a label control object and its child object. The 1st element of the array specifies the status information where the property of the label control object whose string value is "InnerHtml=5/16/99" specified with the line 18 was saved. A property name and "InnerHtml" are equal to status value"5/16/99", and are a "string" type. A semicolon expresses termination of the status value of a given property. As for each condition data value and its related property, a boundary is defined by the termination semicolon when the label control object of a line 18 contains the additional property with which the condition was saved. The zero value shown in the line 19 like the 2nd element of an array shows that the label control object does not contain the child object. Lines 20-27 are the termination tags of the initiation tag which drawing 9 precedes.
[0079] With another operation gestalt, the serialization format called a finite object serialization (Limitted Object Serialization (LOS)) format is used for the condition structure which can be conveyed. Generally, a LOS format specifies the hash table of an identifier / value pair of each control object in which each hash entry contains either of the hash tables of the status information of the property of a control object, and the nest of a child control object.
[0080] A table 2 shows an example of the syntax of a LOS format.
[0081]
[A table 2]

制御オブジェクト	• value type-table$_{opt}$ name-table$_{opt}$	• h<name1;value1>¥t50System.Drawing.Color¥n1BackColor
値	• typed-value • untyped-value • typed-array-value • untyped-array-value • untyped-hashtable-value	• 50<red> • <red> • a50<red;blue;green> • a<red;blue;green> • h<name1;value1>
タイプされた値	• type-ref value-list-start value-ref value-list-end	• 50<red>
タイプされたアレイ値	• array-modifier type-ref value-list-start array-value-ref value-list-end	• a50<red;blue;green>
非タイプ値	• value-ref	• <red>
非タイプアレイ値	• array-modifier value-list-start array-value-ref value-list-end	• a<red;blue;green>
値リスト開始	• <	
値リスト終了	• >	
値リストセパレータ	• ;	
アレイモディファイア	• a	
ハッシュテーブルモディファイア	• h	
非タイプハッシュテーブル値	• hashtable-modifier value-list-start hashtable-value-ref value-list-end	• h<name1;value1>
値の参照	• string-value • bin-ref base64-persisted-object	• ¥"This is a string value.¥"
バイナリ参照	• escape-char b	• ¥b
アレイ値参照	• value$_1$;[value$_2$;[value$_n$]]	• red;blue;green
ハッシュテーブル値の参照	• name-ref$_1$;value$_1$ [value-list-separator name-ref$_2$;value$_2$ [value-list-separator name-ref$_n$;value$_n$]]	• <name1;value1; name2;value2; name3;value3>
名前の参照	• string-name-number • string-name	• 1 • BackColor
タイプの参照	• known-type-number • string-type-number • string-type	• 10 • 50 • System.Drawing.Color

[0082] The following entries are considered as the 1st example of the operation of a LOS format. h<name1;value1;name2;value with \; escaped \> \"characters\"> This specifies the hash table of two pairs of an identifier/values, and all values here are strings. The 1st value is "name1". It is the name to say and is equal to string"value1." The 2nd value is a name of "name2", and is string"value with. ; It is equal to escaped > "characters"."

[0083] The following entries are considered as the 2nd example of the operation of a LOS format. h<control1;h<1;50<blue>;text;hello>control -- 2;h<1;50<red>;control3;> \t50System.Drawing.Color\n -- this specifies the hash table which has the three hash entries "control1", "control2", and "control3" 1 BackColor. The 1st hash entry is a name of "control1", and contains the child hash table which has two entries which are an identifier / value pair. The index "1" for referring to the name of "BackColor" specified at the last of this example is used for the 1st entry name of control1 hash table. The 1st entry value is a "System.Drawing.Color" type which is specified by the index "50" equally to "blue", and is specified near the last of this example. The identifier / value pair of the 2nd entry of control1 hash table contain identifier "text" and value "hello". The hash entry of "control2" contains the child hash table which has one hash entry which uses the index "1" for referring to the name of "BackColor" also here.

This value is equal to "red" and is a "System.Drawing.Color" type. The hash entry of "control3" is empty. Although a hierarchy is specified by the nest of a hash table by the same approach as an above-mentioned operation gestalt, it is thought that the description approach of other hierarchies including the activity of a hierarchy identifier is also within the limits of this invention.

[0084] Drawing 10 shows the process flowchart which loads the status information memorized reception and there in the condition structure in an operation gestalt with this invention which can be conveyed to a control object hierarchy's control object. With the illustrated operation gestalt, unless at least one property of a server side [at least one] control object is changing, the condition structure which can be conveyed is not transmitted from a server (therefore, not returned to a server by the client). Since it is such, it is assumed that the illustrated flow chart exists in the condition structure which at least one element of status information received and which can be conveyed. With another operation gestalt, even if there is no status information in the condition structure which can be conveyed, some which go back and forth between a server and clients will have the condition structure which can be conveyed. With such an operation gestalt, when there is no status information in the condition structure which can be conveyed, decision processing (not shown) can operate so that a status management process may be closed.

[0085] Reception 1000 receives from a client the condition structure which can be conveyed at a HTTP request etc. The marking processing 1002 sets up condition directions of the beginning all the server hierarchy's control objects. The read-out processing 1004 reads the received condition structure which can be conveyed, and extracts status information and a corresponding property type.

[0086] The check processing 1006 performs the integrity check of the condition structure which can be conveyed (using for example, a ##VIEWSTATEMAC value). With an operation gestalt with this invention, it judges whether the;(3) receiving integrity code which is contained in the condition structure which reads a receiving integrity code from the condition structure which can be (1) conveyed, and which can be;(2) conveyed and which calculates the integrity code of itself from status information at least compares with the integrity code which newly calculated, and the condition structure which can be conveyed is altered between round trips of a client in check processing 1006. When the condition structure which can be conveyed is judged to be altered, a server side code can answer by continuing or considering processing without the status information which closes load processing as an error etc. so that this exception may be treated.

[0087] The inspection processing 1008 inspects the file check code contained in the condition structure which can be conveyed. With the operation gestalt of this invention, in the inspection processing 1008 (1) The file check code which carried out;(2); (3) reception which reads the file check code which received from the ##VIEWSTATE field, and which calculates the file check code of itself from the content of the ASP+ resource memorized by the server at least Comparing the newly calculated file check code, the status information of the condition structure which can be conveyed judges whether it corresponds to the same version of the ASP+ resource of a server. It checks not changing between the round trips of the ASP+ resource on a server of the condition structure to a client which can be conveyed using the inspection processing 1008. When the ASP+ resource on a server changes, a server cancels the received status information, and closes load processing, and sends an error to/or a client. With a certain operation gestalt, the control object hierarchy of the first condition is created, and before a server continues the processing to a request, load processing is closed. The traverse processing 1010 traverses a server side control object hierarchy's page control object.

[0088] The syntax-analysis processing 1012 analyzes the syntax of the status information of the control object hierarchy corresponding to the control object of the insert-die ##VIEWSTATE field. The control object which corresponds is a page control object traversed in the traverse processing 1010 1st repeatedly. Repeatedly [of a degree], the control object which corresponds is a descendant control object of a page control object, and the traverse processing 1028 clarifies the location. The syntax-analysis processing 1012 extracts status value, and when analyzed at the beginning, it can change into a given property type the value which was a string format. The load processing 1014 loads the status information the syntax of was analyzed from the ##VIEWSTATE value to the property of a control

object. The directions processing 1016 sets up change condition directions of a current control object. By saving behind only the condition of a control object that the condition changed from the original condition, these directions can be used in order to make into min size of the condition structure which can be conveyed. Or all status information may be saved. It means that change which it changed into the condition of a control object had also set up change condition directions of each control object during handling of postback data, handling of a postback event, and data-coupling processing on a control object hierarchy. However, with an operation gestalt with this invention, any data change directions set up for data-coupling processing are reset so that the condition of the property may not be recorded on the condition structure which can be conveyed, and the size of the condition structure which can be conveyed is further decreased by it. It depends for the condition of such a property on the data specified with the joint relation updated by the data-coupling processing 512 of drawing 5 .

[0089] In the decision processing 1018, when additional status information exists in a current control object, processing progresses to 1022. When it exists, in processing 1022, the syntax of the status information of the following control object property in a current control object is analyzed, and processing progresses to 1014. In the decision processing 1018, when there is no additional status information in a current control object, it progresses to the decision processing 1020, and when other control object status information cannot be found in a ##VIEWSTATE value here, it progresses to processing 1024. In the decision processing 1020, when there is a control object of the addition by which the condition is not processed, it progresses to processing 1026 and the syntax of the status information of the following control object is analyzed from a ##VIEWSTATE value here. In the traverse processing 1028, the following control object corresponding to the new status information the syntax of was analyzed by processing 1026 is traversed. Then, it progresses to the load processing 1014. With an operation gestalt with this invention, repetitive syntax-analysis processing of drawing 10 performs processing in which it does not serialize.

[0090] Drawing 11 shows the process flowchart which saves the status information from a control object hierarchy's control object to the condition structure which is transmitted to the client in an operation gestalt with this invention, and which can be conveyed. The initialization processing 1100 initializes the condition structure which can be conveyed. Computation 1102 calculates a file check code based on the current version of an ASP+ resource. With the operation gestalt of this invention, a file check code is based on the hash algorithms (for example, MD5 etc.) of the content of an ASP+ resource of only hashing the component with which some of contents of the file or files were specified. The load processing 1104 loads the file check code calculated by the condition structure which can be conveyed. The traverse processing 1106 traverses a control object hierarchy's page object.

[0091] In the operation gestalt of this invention, a server side control object can be created using the "MaintainState" property which shows whether the property value of a control object (and the child object) should be saved in the condition structure which can be conveyed. When it is not necessary to save, the traverse processing 1106 can skip a server side control object and its child object in this process. That is not right, and if the condition of a control object should be held according to a "MaintainState" property, the traverse processing 1106 traverses a control object.

[0092] Processing progresses to the read-out processing 1115 by the decision processing 1108. Here, new status value and its type are read from a control object property. Processing 1110 records new status value and its type, and progresses to processing 1111. A property is expressed with the "null" value of the ##VIEWSTATE field when a property does not exist in a current control object for this explanation. If the property of a current control object is still the original condition in processing 1108, it will progress to the decision processing 1111. When another property exists in a current control object in processing 1111, it progresses to the decision processing 1108.

[0093] When other properties do not exist in a current control object in processing 1111, it progresses to the decision processing 1112, and when another control object exists in a hierarchy, it progresses to the traverse processing 1114 here. The traverse processing 1114 traverses the following control object in a control object hierarchy, progresses to the decision processing 1108, and accesses the property of a control object. In processing 1112, when other control objects do not exist in a hierarchy, the new status

information (a status value type is included) acquired by previous processing is serialized, and it progresses to the serialization processing 1116 which memorizes them in the condition structure which can be conveyed. Processing 1118 calculates an integrity (using MD5 algorithm) code, and processing 1120 memorizes an integrity code in the condition structure which can be conveyed for example, in the ##VIEWSTATEMAC field. Integrity code processing and field check processing are arbitrary, and can be excluded with another operation gestalt of this invention. Processing 1122 transmits the condition structure which can be conveyed to a client at the HTTP request containing the HTML code which was generated by for example, the control object hierarchy and by which the rendering was carried out etc.
[0094] The operation gestalt of this invention in this description is performed as a logic step of one or more computer systems. Logic processing of this invention is performed as a machine module with which it interconnected within the computer system beyond (2) 1 ** as a processor execute step sequence by which executive operation is carried out by the computer system beyond (1) 1 **. Activation is the problem of selection and it depends for it on the performance requirements of the computer system which performs this invention. Therefore, the logic processing which constitutes the operation gestalt of this invention of a publication on these descriptions can be variously expressed as processing, a step, an object, or a module.
[0095] A description, an above-mentioned example, and above-mentioned data offer the structure of the operation gestalt of this invention, and perfect explanation of an activity. Since this invention can be carried out with many gestalten, without deviating from the pneuma and the range of this invention, this invention is in an attached claim.

[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]
[Drawing 1] The web server which generates dynamically the web page contents displayed on the client in an operation gestalt with this invention is shown.
[Drawing 2] The flow chart of processing for processing of a client side user interface element and a rendering is shown using the server side control object in an operation gestalt with this invention.
[Drawing 3] An example of the module of a web server used with an operation gestalt with this invention is shown.
[Drawing 4] An example of a useful system is shown in performing an operation gestalt with this invention.
[Drawing 5] The process flowchart showing processing of the page object in an operation gestalt with this invention is shown.
[Drawing 6] An example of the part of the dynamic contents file (for example, ASP+ resource) in an operation gestalt with this invention is shown.
[Drawing 7] According to an example of the part of the dynamic contents file of drawing 6 , as a result of being generated by one or more server side control objects, a code is shown.
[Drawing 8] The control object hierarchy corresponding to the ASP+ resource of drawing 6 and the ##VIEWSTATE field value of drawing 7 are shown.
[Drawing 9] The nest version of the ##VIEWSTATE field value of drawing 7 is shown.
[Drawing 10] The process flowchart which loads the status information in which the condition structure in an operation gestalt with this invention which can be conveyed was stored by reception and it to a control object hierarchy's control object is shown.
[Drawing 11] The process flowchart which saves the status information from a control object hierarchy's control object in an operation gestalt with this invention to the condition structure which can be conveyed, and transmits to a client is shown.
[Description of Notations]

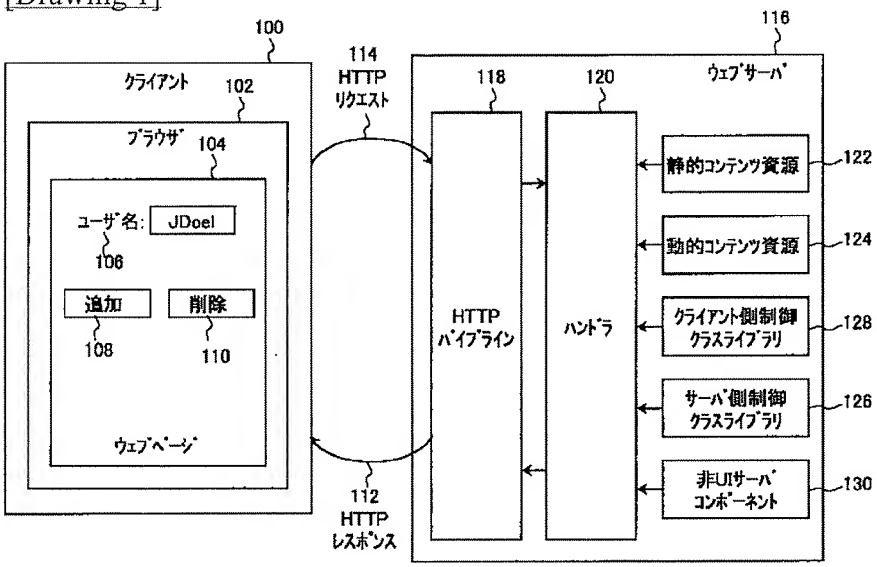
[Translation done.]

* NOTICES *

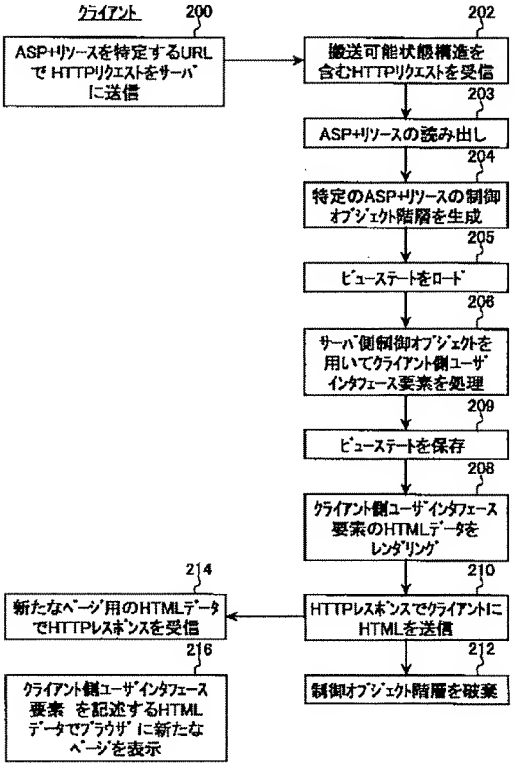
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

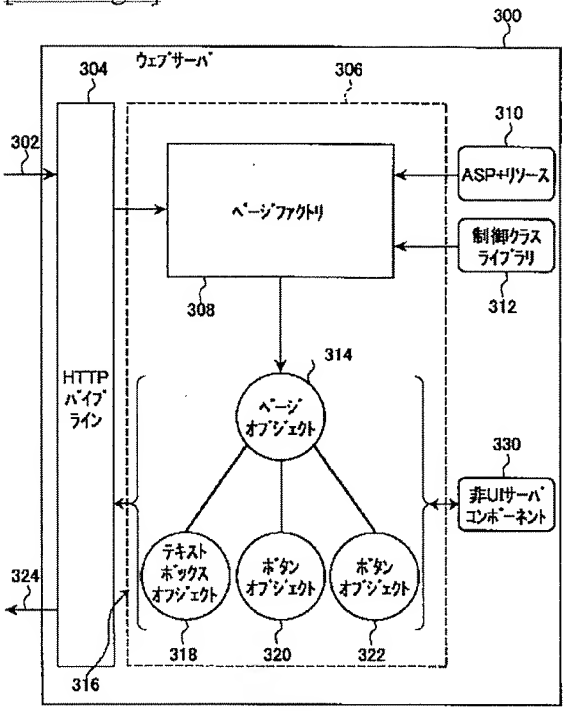
[Drawing 1]



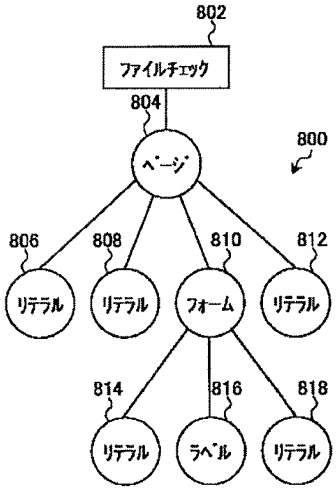
[Drawing 2]



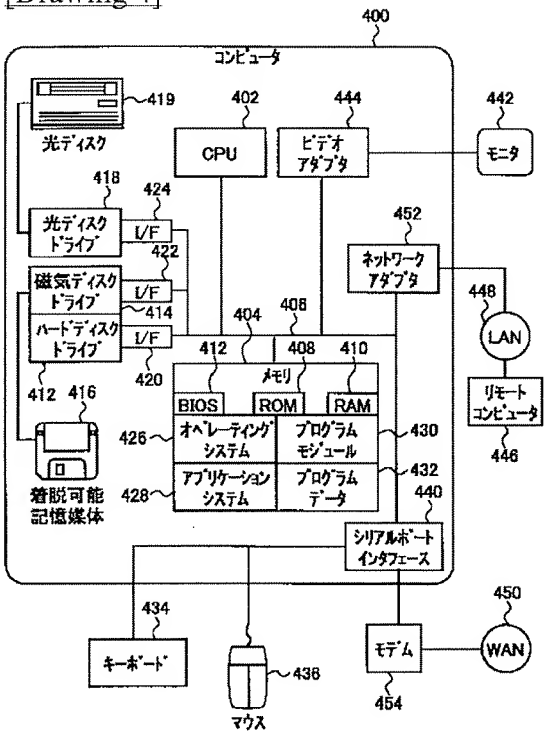
[Drawing 3]



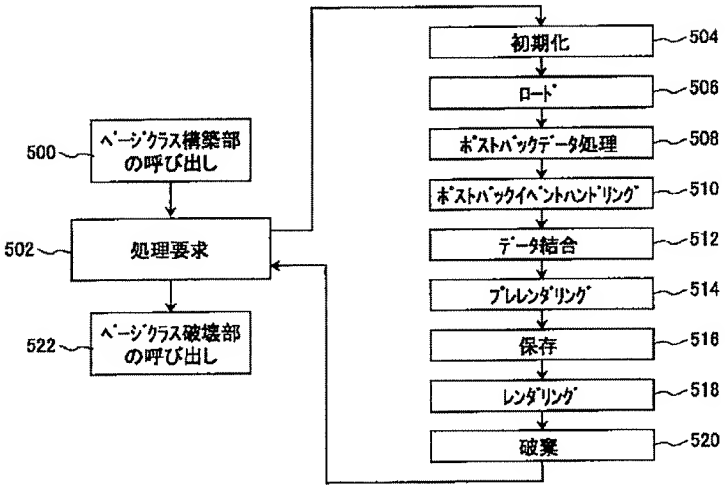
[Drawing 8]



[Drawing 4]



[Drawing 5]



[Drawing 6]

600

```
1    <html>
2        <script runat=server>
3            Overrides Sub Load()
4                If (IsFirstLoad) Then
5                    Message.Text = "You've never visited this page before!"
6                Else
7                    Message.Text = "Last visited: "& State("LastAccessed")
8                End If
9            Overrides Sub Save()
10               State("LastAccessed") = Now
11           End Sub
12       </script>

13       <body>
14           <form runat=server>
15               <span id="Message" runat=server/>
16           </form>
17       </body>
18   </html>
```

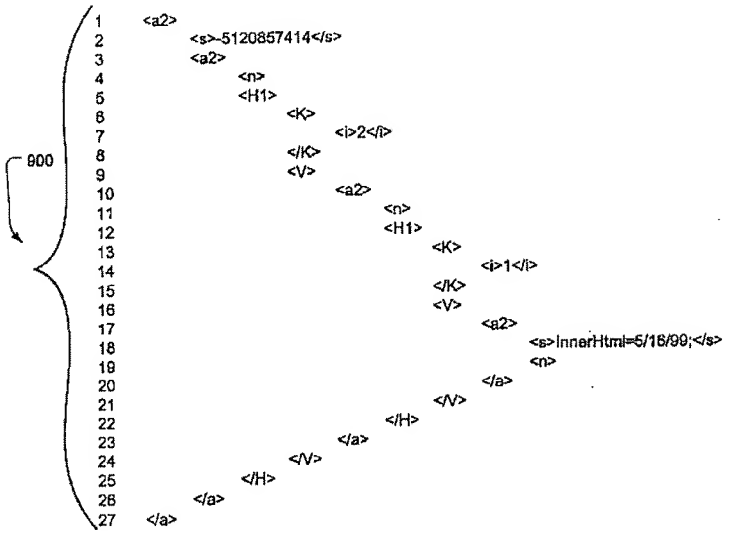
[Drawing 7]

700

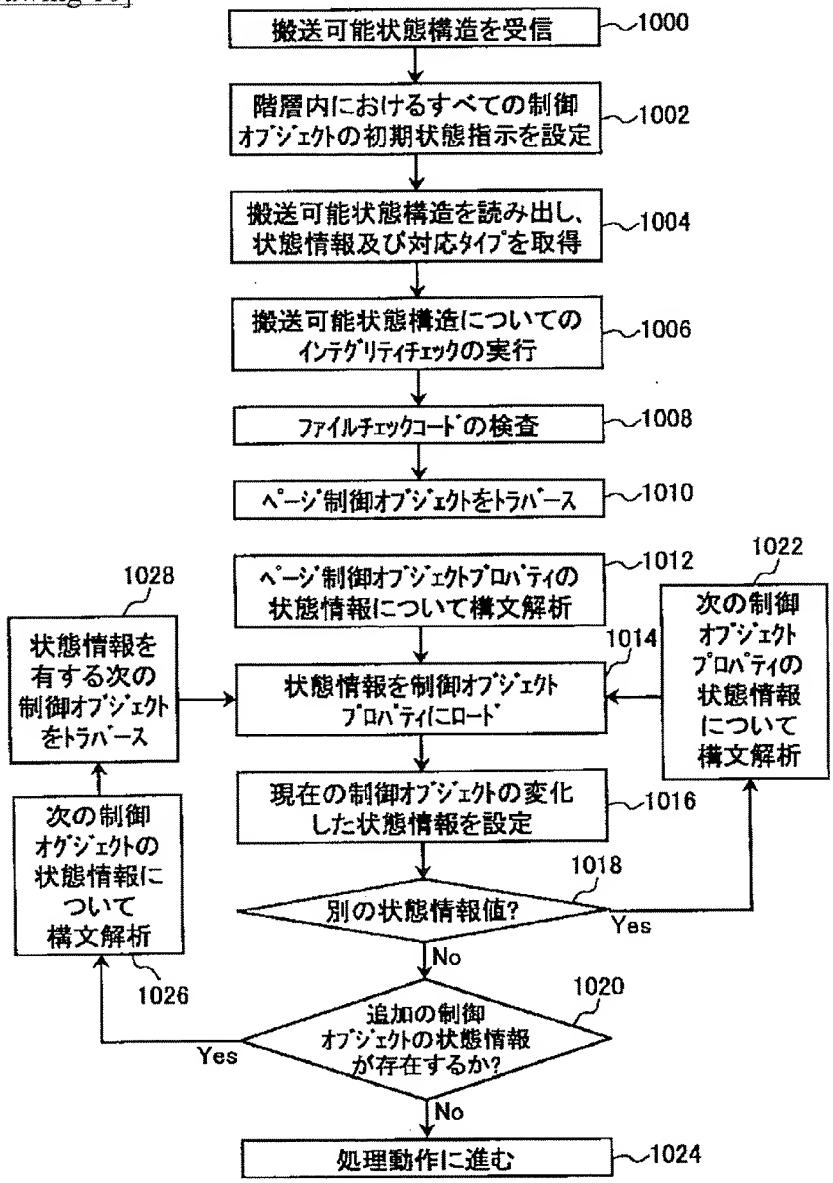
```
1    <form>
2        <input type=hidden name="__VIEWSTATE"value="<a2><s>-5120857414</s><a2>
     <n><H1><K><i>2</i></K><V><a2><n><H1><K><i>1</i></K><V><a2><s>InnerHtml=5/16/99;
     </s><n></a></V></H></a></V></H></a></a>">
3        <input type=hidden name ="__VIEWSTATEMAC" value="43434333433">

4        <span id="Message">Last Post Back: 5/6/99</span>
5    </form>
```

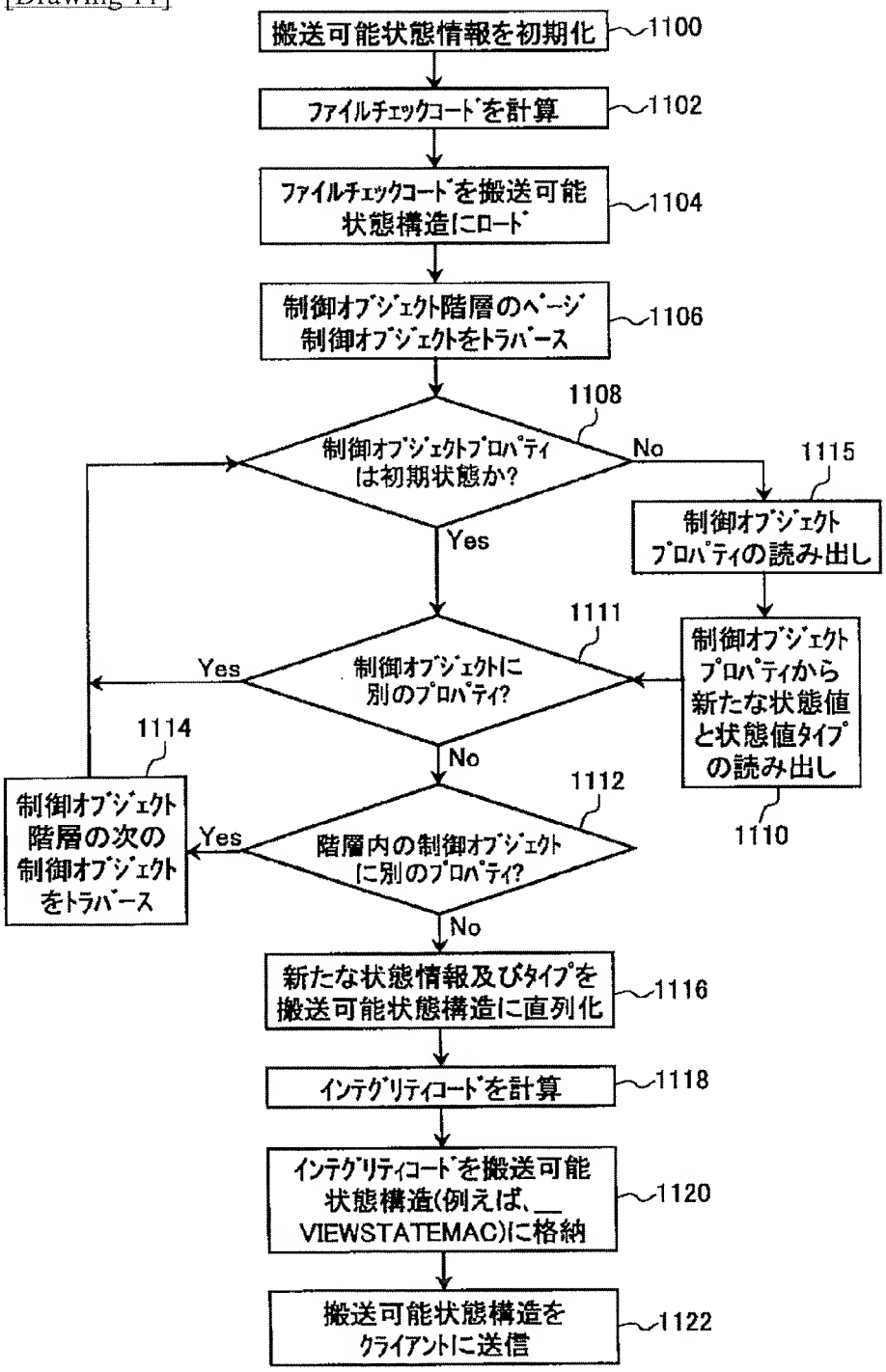
[Drawing 9]


900
1 <a2>
2 <s>-5120857414</s>
3 <a2>
4 <n>
5 <H1>
6 <K>
7 2
8 </K>
9 <V>
10 <a2>
11 <n>
12 <H1>
13 <K>
14 1
15 </K>
16 <V>
17 <a2>
18 <s>InnerHtml=5/16/99;</s>
19 <n>
20
21 </V>
22 </H>
23
24 </V>
25 </H>
26
27

[Drawing 10]


搬送可能状態構造を受信
1000
階層内におけるすべての制御オブジェクトの初期状態指示を設定
1002
搬送可能状態構造を読み出し、状態情報及び対応タイプを取得
1004
搬送可能状態構造についてのインテグリティチェックの実行
1006
ファイルチェックコードの検査
1008
ページ制御オブジェクトをトラバース
1010
ページ制御オブジェクトプロパティの状態情報について構文解析
1012
1022
次の制御オブジェクトプロパティの状態情報について構文解析
1028
状態情報を有する次の制御オブジェクトをトラバース
1014
状態情報を制御オブジェクトプロパティにロード
現在の制御オブジェクトの変化した状態情報を設定
1016
1018
別の状態情報値?
Yes
No
次の制御オブジェクトの状態情報について構文解析
1026
1020
追加の制御オブジェクトの状態情報が存在するか?
Yes
No
処理動作に進む
1024

[Drawing 11]



[Translation done.]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

WRITTEN AMENDMENT

---------- [procedure amendment]
[Filing Date] June 12, Heisei 13 (2001. 6.12)
[Procedure amendment 1]
[Document to be Amended] Description
[Item(s) to be Amended] 0096
[Method of Amendment] Modification
[Proposed Amendment]
[0096]
[Effect of the Invention] According to this invention, in the client / server model which does not perform status management, it is not held between two HTTP requests of the status information of a server side control object at a server side as mentioned above. By the suitable server side control object from between a client and a server being conveyed with the condition structure which can be conveyed The client side user interface element displayed on a web page can be processed and generated, and dynamic contents become possible [developing the target web page more easily].
[Procedure amendment 2]
[Document to be Amended] Description
[Item(s) to be Amended] Explanation of a sign
[Method of Amendment] Modification
[Proposed Amendment]
[Description of Notations]
100 Client
102 Browser
104 Web Page
106 User Name
108 Addition
110 Deletion
112 HTTP Response
114 HTTP Request
116,300 Web server
118 304 HTTP pipeline
120 Handler
122 Static Contents Resource
124 Dynamic Contents Resource
126 Server Side Control Class Library
128 Client Side Control Class Library
130 330 Non-UI server component
308 Page Factory

310 ASP+ Resource
312 Control Class Library
314 Page Object
318 Text Box Object
320 322 Carbon button object
400 Computer
404 Memory
412 Hard Disk Drive
414 Magnetic Disk Drive
416 Removable Storage
418 Optical Disk Drive
419 Optical Disk
420, 422, 424 I/F
426 Operating System
428 Application Program
430 Program Module
432 Program Data
434 Keyboard
436 Mouse
440 Serial Port Interface
442 Monitor
446 Remote Computer
452 Network Adaptor
454 Modem
802 File Check
804 Page
806, 808, 812, 814, 818 Literal
810 Form
816 Label

[Translation done.]